Chapter 2
Active Learning in Large Lectures

Diane Kitchin

Abstract An increasingly diverse student body combined with pressures to demonstrate excellence in teaching and improve results presents challenges for lecturers. Active learning techniques have attracted interest and discussion amongst educationalists. This chapter investigates the challenges and gives a practical guide to techniques that have been used effectively in a large lecture situation with first year students.

Keywords Active learning · Large lectures · Computing education
Constructivism

2.1 Introduction

Changes in higher education over recent years have seen a rise in the number of students going to University and consequently a more diverse student body. This has led to a need to help students make a smooth transition between school and University and to adjust to different environments, different delivery styles and different expectations. With increased competition for students amongst Universities, and the new demands of the Teaching Excellence Framework (TEF), comes increased expectations on lecturers—we must ensure our students pass our modules, and moreover pass them with good marks. Excellent and effective teaching is seen as a key factor by Universities in attracting students to their courses. The University of Huddersfield mentions the quality of its teaching in four out of six of the factors that demonstrate its excellence on its 'About us' page for new and prospective students. (University of Huddersfield 2018). Active learning pedagogies was one of the themes identified by the Higher Education Academy (HEA) in its review of the written submissions Universities included as part of their TEF documentation, in support of the assessment criteria for Teaching quality, Learning environment and Student and outcomes learning

D. Kitchin (✉)
University of Huddersfield, Huddersfield, UK
e-mail: d.kitchin@hud.ac.uk

J. Carter et al. (eds.), *Higher Education Computer Science*,
https://doi.org/10.1007/978-3-319-98590-9_2

gain. (HEA 2017). In the HEA report on TEF 2 the authors state that "Course design comes out as a prevalent aspect for providers upgraded to a Gold award …. Features mentioned in some statements included: … use of active learning…" (Moore et al. 2017).

Additional challenges to providing an active learning experience are that group sizes are often large, particularly for first year teaching, where lectures may be delivered to groups of 150 students or more. At Huddersfield with this diversity in the student body we have seen an increase in the number and level of student support initiatives. We have also seen a shift in the type of learning activities available, with more studio work, project work and group work. Research in approaches to learning advocates a more learner-focused approach in teaching, with active learning being much-discussed over recent years.

In this chapter we focus on the practical techniques adopted to overcome these challenges and foster an active learning environment, particularly in a large lecture situation. The remainder of this chapter is structured as follows: Sect. 2.2 discusses the challenges and motivation for this work in more detail; Sect. 2.3 describes active learning and reviews some of the literature; Sect. 2.4 describes the specific, practical techniques used to deliver active learning in a large Computing Science and Mathematics lecture; Sect. 2.5 reflects on outcomes and the use of these techniques and looks at possible further development and Sect. 2.6 gives a summary and presents our conclusions.

2.2 Challenges and Motivation

The specific context for this discussion is the delivery of a Computing Science and Mathematics module to first year students. This is a long-running module, which has existed in some form or another for many years, and which has been delivered by me for almost 10 years. It covers topics such as set theory, graphs and trees, propositional logic, sorting algorithms, Finite state automata, grammars and languages, regular expressions, binary search trees and tree traversal algorithms. It is typically taken by a very large group of around 100–170 students. Invariably, there will be a very wide range of abilities in each group. Some students won't have done any Maths since GCSE two or more years previously and may only have a grade C. Others will be very able students who have studied A-level Maths or Computing or both, and so they may have covered some of the topics already. There will also be a small number of mature students, who could have been out of formal education for a number of years, and who may be nervous about the subject. There will also be a number of international students, who may have had very different educational experiences. Again some of them will have covered similar material before, while others may not. A large mixed-background, mixed-ability group presents many challenges. How can we keep the attention of and give new challenges to students who are familiar with a topic, whilst not overwhelming and alienating students who view the material as difficult and possibly scary?

There are a number of barriers to understanding in such a group. The cultural backgrounds and life experiences of students can vary widely, so I always try to bear in mind that not all examples will be appropriate for such a mixed audience. For example, once when teaching Java classes and inheritance to a mixed group of students, using what seemed to be a simple Bank Account example, I was puzzled as to why one of the international students was struggling to understand the material. It turned out to be because he was unfamiliar with the meaning of terms like Current Account, Savings account etc., rather than the programming language concepts. Some students may have been taught just to accept and learn what they are told, and will not have been encouraged to ask questions. Also many students just assume that maths is difficult or that they can't do it, and often don't recognise the difference between the general maths they studied at school and the discrete maths relevant to computing. With such a large and diverse group, it is important to keep all the students interested and engaged, and to make sure that the teaching materials don't provide an unnecessary barrier to learning or exclude any students. Even small things can make a difference—such as not choosing exclusively male or white British names in examples. So a simple set theory example of a set of students = {Bob, Tom, Harry} could make a student feel excluded if the examples all follow a similar pattern.

2.3 Active Learning

In the past, traditional lectures were often viewed merely as a vehicle to convey a large amount of information to a lot of students at one time. Students were expected to be largely passive, just listening and taking notes, and perhaps occasionally asking or answering questions. Research has shown that audience attention in lectures begins to wane every 10–20 min. This traditional approach has been questioned by educationalists and researchers, with theories such as Constructivism (Bruner 1960, Piaget 1950, and others) advocating active learning.

Constructivist theories see learning as an active process, whereby students build on and adapt existing knowledge or knowledge structures in the mind known as 'schemata', so developing and deepening their understanding. Knowledge isn't something that exists in isolation, it requires a context and connections to what we already know, so when we learn we are not just adding new information, we also have to make sense of it and give it meaning in relation to existing knowledge. "Learning is thus an active process of individual transformation and changes in understanding" (Fry et al. 2015, p. 65). Acceptance of this theory entails a change in the way teachers think about teaching and how they do it. Constructivists argue that the teachers must encourage a deep approach to learning, with context and expectations being set by the teacher. In addition, some researchers emphasise the importance of learning outcomes, arguing that the curriculum should be constructively aligned (Biggs and Tang 2009) with the teaching environment and assessment, and that students should be made aware of and understand these learning outcomes.

The question now arises “what exactly is active learning?”. There are many different definitions in the literature. The term active learning originates in the work of Revans (1971). Prince (2004) offers a simple and useful explanation, defining “the core elements of active learning to be introducing activities into the traditional lecture and promoting student engagement”.

Bonwell and Eison (1991) define active learning as “instructional activities involving students in doing things and thinking about what they are doing.” According to Weltman (2007) the activities can range from the very active, such as physically making something to doing something like playing a game or to the less-involved such as watching a video, and then applying what has been learned from this in some way. A commonly used lower-level example of active learning could just be allowing time in a lecture for students to think about or discuss the material presented. The argument is that since students can only concentrate effectively for about 15 min, introducing pauses two or three times in a lecture, or having some change of activity or focus is beneficial and a simple way of making the learning experience more active. Other commonly mentioned methods include problem based learning, and cooperative and collaborative learning.

We should also ask, before adopting an active learning approach, what evidence there is for its effectiveness. Weltman (ibid.) cites various studies (Raelin and Coghlan 2006; Sarason and Banbury 2004; Sutherland and Bonwell 1996; Umble and Umble 2004) which have found supporting evidence for the effectiveness of active learning techniques. According to a review by Prince (2004) there is evidence in favour of active learning. He cites Bonwell and Eison (1991) who “conclude that it leads to better student attitudes and improvements in students’ thinking and writing”. His review includes the work of Felder et al. (2002) who recommend active learning as “one of the teaching methods that work”. However, Prince also points out that the support for active learning from the literature he reviews is not conclusive and the improvements may only be small. Prince also reviews empirical support for active learning and concludes that “introducing activity into lectures can significantly improve recall of information while extensive evidence supports the benefits of student engagement” (Prince 2004, p. 4).

2.4 Techniques and Practices

As previously mentioned, the techniques, practices and examples given in this section are all taken from a first year Computing Science and Mathematics module. The material used has been built up over a number of years by colleagues who taught the module previously (Ron Simpson, Barbara Smith, Lee McCluskey and John Turner) and myself. A lot of the material is theoretical, technical and factual, with little room for individual interpretation. Nevertheless, students have to build up their own understanding and mental models. So the aim is to make the lectures a mixture of factual delivery combined with various active learning techniques, to encourage engagement and facilitate learning.

The active learning techniques which are used are:

a. Motivation with real applications
b. Worked examples on a Visualiser
c. Students work on further examples in the lecture
d. Animations and applets
e. Short videos
f. Active participation and enactment.

Examples of the use of all of these techniques will be given in the remainder of this section.

Often in large lectures students are largely passive—just listening and perhaps taking notes. From the start of the term students are encouraged to be active learners in these lectures, participating in exercises and even activities. They should not think of themselves as empty vessels passively waiting to be filled with information and knowledge, but as active participants in a process. So it is made clear from the first week that they will be expected to take an active part in lectures, to bring pen and paper each week and to ask and answer questions.

Some students will see Maths as difficult, or a topic that they have previously found challenging, and some won't have done any computing science at all. They often don't initially see the relevance of theoretical material to their studies—"what has maths got to do with computing?" they may ask. It is important to build their confidence and to give them some motivation for learning what they may perceive to be hard and unnecessary. Thus a good starting point for every lecture is an example of a practical application within the real world or from computing of the particular topic to be studied that week. It also helps give some context for what may be completely new material. This, in terms of constructivist theory, helps them to begin to build or adapt a knowledge structure in their minds. For example, few students will have studied graph theory and will not link it with computing. Some concrete examples of the application of graph theory that could be given would be for modelling a network configuration or modelling dynamic processes, while trees are commonly used data structures they will meet in computing. For Finite state machines (FSM) a computer game would make a good example—where the states of the machine represent the states of a games, something almost all student will be familiar with. The states represented could be "explore", "attack", "evade" and so on with transitions between states labelled "player nearby", "no player in sight", "player attacks", "player idle" etc.

Once the context, motivation and applications are clear, these examples can then be followed by some definitions of the terminology or concepts being introduced. Wherever possible each concept or item should be accompanied by an example or diagram—e.g. vertex, edge, graph, tree. As with all written examples the Visualiser is used for this. In logic a simple example of a proposition is given—maybe something intriguing or amusing—e.g. an example of a simple proposition could be: "My real name is Trillian" (from Hitchhiker's Guide to the Galaxy), or perhaps something more current, but the idea is to choose something that is familiar to them from the everyday world. The students can then be asked to think of and write down their own example, perhaps something relating to them, and either share it with the person next to them or tell the whole group, when asked.

The general pattern followed is exposition, example, activity, so, as soon as possible the students are given a small example or problem to work out. For example, when teaching set theory, after the concept has been explained and an everyday example given, they can then be asked to raise their hand if they are in the set of students with brown hair. Then the set of students with blue eyes, then the intersection (i.e. if they are in both sets) and so on. Another everyday example used is something like the "Who am I game" where students are shown a collection of pictures of men and women, with and without hats, glasses and beards and asked to determine the members of various sets, and the results of applying simple set operators. The idea is that the theory is more likely to be retained if they've been physically involved by putting their hand up, or writing a simple sentence etc.

As the lecture progresses the examples and exercises become increasingly technical. After introduction of all basic definitions and concepts I do a simple worked example with the students. I prefer to use a Visualiser for this rather than writing on a white board, because it is clearer for the students to see, as it is projected on to the main screen, and also I can still look at and talk to them while doing the example, rather than turning my back to them.

Two or three problems of increasing difficulty may be given for the students to work on. For example, three sentences in English to be translated into propositional logic, or using inference rules to prove logical arguments. I always do at least 1 or 2 problems with them first and then let them have a go on their own. Alternatively, if the exercise is more difficult or time-consuming, such as a more complex FSM, then I would talk through it, but involving the students in its construction, asking them what to do next at each step—e.g. which edge or label should be added next, where it should go etc., so continually encouraging involvement and engagement, and discussing why a particular suggestion might or might not work. I might ask "would adding a particular transition as a loop allow the generation of invalid strings, which would mean we should add a new state instead"? It is important to keep checking for understanding—encouraging a culture of asking questions (I say I don't expect them to understand everything immediately and so I'm worried if they're not asking questions).

For the more able students, hopefully the lecture serves as a reminder or refresher if they have covered the material before, and also a way to help them reinforce or add to what they already know. More challenging questions can be included in the tutorial exercises. These students often are more confident in answering questions or volunteering information in the lecture, which can help weaker students who might be reluctant to speak up.

A lecture on recursion will now be used as an example which incorporates many of the active learning techniques mentioned at the start of this section. The learning outcome is that students should develop an understanding of the working of a recursive algorithm—which then leads on in following weeks to understanding different recursive algorithms used in sorting or tree traversal. It begins with an everyday example that should build on existing knowledge. The context is them widened with examples from computing and maths to help motivate further learning. Eventually there is a physical enactment of the algorithm by the some of the student followed by a more formal worked example.

The example from everyday life to introduce the idea of recursion is walking across a moor in the fog. If we just try to find our way across the foggy moor by attempting to walk in a straight line in one direction, we can easily get lost and end up walking round in circles. However, if we use our compass to identify a small feature or landmark about 100 m ahead in the right direction, such as a rock or hillock, which we then walk to, and keep following this strategy, we have broken our problem down into a smaller one which we can solve easily (walking 100 m). We can keep applying this method to the smaller problem that remains, as we are now 100 m close to our destination. We can write this as an algorithm for crossing the rest of the moor:

- If you can see your destination, walk to it;
- If not,
  - choose a landmark in the right direction 100 m further on and walk to it;
  - cross the rest of the moor.

This is a recursive algorithm; we have defined crossing the rest of the moor in terms of… crossing the rest of the moor.

A useful way to change the focus of student attention is to use short animations or applets within a lecture to illustrate a particular concept. This lecture next moves on to another example where recursion can be used—the Tower of Hanoi. There is a dynamic application showing the Tower of Hanoi (Dynamic Drive). This puzzle has three pegs and a number of discs of graduated sizes, each with a hole in the middle so that they can be placed on a peg. The aim is to move the tower of discs from one peg to another. Although we could physically move them in one go, this is not allowed by the rules of the puzzle:

- We can only move one disc at a time.
- We can only put a disc on top of a larger disc, or on the base board.

This puzzle according to legend, originates in a monastery in Hanoi (Vietnam), where the monks are engaged in moving a tower of 64 discs from one peg to another; when they have finished, the legend says that the world will end.

The application allows the user to select the number of starting discs, and then to move them one at a time from one peg to another. It also shows the minimum number of moves and the number of moves made by the user. I usually run this starting with three discs, moving them from the start peg to the finish peg, and then increase the number of discs by one on each attempt, trying to keep to the minimum number of moves. This can be quite stressful for the lecturer, once the number of discs increases, but you may be rewarded with a round of applause if you can do a sufficiently impressive number of discs without making a mistake—six is a good number to aim for! The students can also have a go at this themselves after the lecture or in one of the tutorials, to help increase engagement with the topic.

The demonstration can then be followed by showing the pseudocode for a recursive algorithm to solve the problem.

So far we have moved from a simple everyday example, followed by an outline recursive algorithm, to a demonstration, followed by a more detailed recursive algorithm. The next step is to ask students to take part in animating a more mathematical example of a recursive algorithm. The problem to be solved is calculating the factorial of an integer—8!

I explain that I ask a student (A) to work out 8! for me. He's not keen on multiplying, so this seems like a lot of work, but he can see that 8! is the same as 8 * 7! He thinks that multiplying something by 8 would be manageable, so he asks a friend to work out 7! for him. The friend (B) feels the same way about multiplying as A, so she asks C to work out 6! for her, and she knows that all she has to do is multiply C's answer by 7. They carry on in this way: C asks D for 5!, D asks E for 4!, E asks F for 3!, F asks G for 2!, and G asks H for 1!

But finally, 1! doesn't need any multiplying, so H knows that the answer is just 1, and passes this back to G. Now G has to multiply that answer by 2, and passes the answer (2) to F, who multiplies that by 3, and passes her answer (6) to E. E multiplies 6 by 4 and passes 24 to D; D multiplies 24 by 5 and passes 120 to C, C multiplies 120 by 6 and passes 720 to B; B multiplies 720 by 7 and passes 5040 to A, who finally realises that multiplying by 8 isn't necessarily that easy but does it anyway and gets 40,320 to pass back to me.

I have pre-prepared A4 cards with the relevant numbers and calculations written on them. i.e. "Calculate 8!", "Calculate 7! and so on. I ask for 8 student volunteers and ask them to stand at the front of the lecture theatre. These cards are given to the students and we begin the animation of the algorithm, with each student passing the relevant Calculate card to the next in turn. When we reach the base case of 1!, the last student then hands back a pre-prepared card with the answer—i.e. 1! = 1. The next student in line takes this, and then hands on a card saying 2! = 2 * 1 = 2, and so on until finally card with the answer is passed back to me.

Following this change of focus and the physical activity, the students are then shown and talked through some Java code to calculate the factorial of an integer. After explaining the code, an example (5!) is drawn using the Visualiser, showing each recursive call until the base case is reached, and then the unwinding of the recursion until the answer is returned. All worked examples done on the Visualiser are made available after the lecture via the University VLE, along with the lecture notes and tutorial exercises. Lecture capture software also enables students to watch all or selected parts of the lecture again, facilitating further consolidation and reflection.

There are many animations and videos available; some useful examples follow. When teaching sorting algorithms there is an interesting animation from Toptal (n.d.) that gives a visual representation of several different algorithms sorting a number of lines of different lengths in real time, so that students can easily see the differences in performance, given different initial conditions. Also the Hungarian folk dancing representation of Quick sort on YouTube can give a useful break and change in focus and activity, and inject some humour into the lecture.

One of the key techniques is allowing students time in the lectures to do short exercises to reinforce what has been taught. For example, in propositional logic, after explaining how to translate from natural language sentences into propositional calculus, they will be given three small problems to solve. The first one I will do with them on the Visualiser, and I will then ask them to do the second problem themselves on paper. Students can either work alone or in pairs, allowing an opportunity for reflection and collaborative work. After a few minutes we will talk through the solution, and then the process is repeated for the third problem. These simple exercises allow the students to reflect on and apply the material presented to them. It can also highlight any areas they have not understood, so they can be encouraged to ask further questions as a result of attempting the exercises.

All lectures have an associated tutorial, where students can reinforce their learning and apply it to graded exercises. There is also the opportunity for collaborative work if they choose. The aim is for this knowledge to be useful to and used by the student in the rest of their University course and in their working life.

2.5 Outcomes and Reflection

How do we assess whether these methods work? Prince (2004, p. 2) suggests that the outcomes that should be considered include "measures of factual knowledge, relevant skills and student attitudes, and …student retention".

We can look at the outcomes of the assessment, which for this module is a 2-hour examination. We can look at Lecture capture star ratings and module evaluations. We can listen to feedback in student panels. From my own work I have noted that those lectures made available via lecture capture that get a higher star rating are the ones I've discussed here—that is propositional logic, recursion and sorting algorithms, which have the highest level of active learning content. Anecdotally, students on the whole are happy with the teaching they receive in this module as evidenced by their comments in our Student Panel meetings.

While the outcomes mentioned have all been good, they don't tell us which particular aspects of the teaching are the most effective. They don't tell us what may in fact work better, and they don't tell us which concepts students find the most challenging.

From my own experience, it seems that some students respond better than others to the active learning techniques discussed. For example, when asked to work on examples in the lecture, some students will work enthusiastically on the problem, discussing it and sharing solutions with their neighbours. Others are less willing to join in—perhaps finding the problem daunting or difficult. Doing a couple of exercises with them first can help, and I also try to give them an easy way to start—so, for example, when drawing a FSM, I suggest they think of the shortest possible legal input, and draw a diagram for that first. This may be as far as they get, but hopefully they get some sense of achievement from this. The worked solution then shows them how to build on this.

Videos and animations are generally well-received, and seem to hold everyone's attention. Obviously they have to be carefully chosen to be both interesting and informative. For the student recursion exercise it is useful to mention it to a few students beforehand, in one of the tutorials perhaps and check they will be willing to volunteer on the day, otherwise it can take a little while to encourage students to stand at the front. However, once everything is in place, it engages the audience.

We might want to consider which students benefit the most. All students should benefit to some extent from being more engaged with what is going on—active learning should be more interesting than just sitting there. Students who don't benefit tend to be those who haven't engaged with the course or the module, or those who have pre-conceived ideas about theory and maths, and believe that they can't do it. Additionally, students who have been absent and not caught up or are finding the module difficult may find the in-lecture exercises too hard. Extra optional support sessions are provided on a weekly drop-in basis to help with this, but it is difficult to ensure that all the students who need them attend. There will always be some students whose reaction to something being difficult is to ignore it and stop attending. Some very able students may find the exercises too easy, but can be given additional advanced exercises to do in the tutorials or in their own time. Ideally for all students their mental knowledge structures should continue to develop as they build on what they have learned in the rest of their degree course. The literature provides some support for the effectiveness of active learning, although many reviewers have identified a need for further study. The conclusions given by Prince (2004, p. 7) are that there is "support for all forms of active learning examined …students will remember more content if brief activities are introduced to the lecture".

What pitfalls are there in adopting an active learning approach? There is a risk that if the exercises are too difficult then the students will just use it as an opportunity to chat amongst themselves. The amount of time given to work on an exercise should be limited, and the lecturer should intervene and go through the example if it seems that students are struggling or not working effectively. It can be difficult to gauge what is the correct amount of time to allow to complete an exercise, as not all students work at the same pace. But by keeping a close eye on who is doing what, who is writing, and general noise levels, it can be done.

It is also important to set the right initial tone and atmosphere as far as possible. So normally students are expected to listen and not chat or disturb other students—but when doing an exercise, they can discuss things amongst themselves. However, once the lecturer beings going through the answer, then their full attention should be expected. Thus changes in activity are accompanied by changes in pace, tone, atmosphere and noise levels. Technology can also cause problems—so any links to videos or animations should be checked before the lecture, in case they no longer work.

It might be asked if content has to be reduced to incorporate active learning. Over the years that this module has been delivered, some topics have been removed through natural wastage, for example, because they no longer feed into specific second year modules—so the removal of a second year Formal Methods module resulted in material on relations and functions being omitted. Such changes in content allow for an increase in the amount of active learning that can be included. Material can also be restructured and reworked to allow for the inclusion of active learning material—perhaps replacing a wordy explanation with a shorter one accompanied by a video or animation, for example.

Finally, if we want to extend the use of active learning techniques, then further development could include the use of a Student Response system for students to give answers to questions to assess their level of understanding. This might help make it clearer to the lecturer which specific aspects or topics students are struggling with. A system developed at Huddersfield enables teachers to initiate questions, and students to respond using their own mobile devices such as phones, laptops or tablets. It allows data to be collected and automatically stored for future retrieval (Meng and Lu 2011). The system can be used in activity-based or problem-based educational settings regardless of the size, age or knowledge background of the learning group. This could provide useful information as to where additional explanation or examples would be beneficial, and help better assess the usefulness of the active components, as well as actually providing an additional active component in the lecture.

2.6 Summary and Conclusion

This chapter has discussed the challenges of teaching large, diverse groups of students, it has described active learning, looked at the motivations for its use and reviewed some of the literature. We have presented the specific, practical techniques and examples used to deliver active learning in a large Computing Science and Mathematics lecture and then considered the outcomes and potential pitfalls of using these techniques along with possible further developments.

References

Biggs J, Tang C (2009) Teaching for quality learning at university, 3rd edn. Open University Press, Maidenhead

Bonwell CC, Eison JA (1991) Active learning: creating excitement in the classroom, ASHEERIC Higher Education Report No. 1, George Washington University, Washington DC

Bruner J (1960) The process of education Cambridge. Harvard University Press, MA

Dynamic Drive (no date) The Tower of Hanoi. Retrieved from http://www.dynamicdrive.com/dynamicindex12/towerhanoi.htm

Felder R, Brent R, Stice J (2002) National effective teaching institute: workshop materials. American society for engineering education annual conference, Montreal, Quebec, Canada

Fry H, Ketteridge S, Marshall S (2015) A handbook for teaching and learning in higher education: enhancing academic practice. Milton Park, Abingdon, Oxon, Routledge

HEA (2017) TEF2 Infographic. Retrieved from https://www.heacademy.ac.uk/knowledge-hub/evidencing-teaching-excellence

Meng Z, Lu J (2011) Opportunities of interactive learning systems with evolutions in mobile devices: a case study. In: Proceedings of the 2011 international conference on internet computing ICOMP 2011. CSREA Press, pp 238–244. ISBN 1601321864
Moore J, Higham L, J Sanders J (ARC Network) in partnership with Jones S, Candarli D, Mountford-Zimdars A (2017) Evidencing teaching excellence analysis of the teaching excellence framework (TEF2) provider submissions. Retrieved from https://www.heacademy.ac.uk/knowledge-hub/evidencing-teaching-excellence
Piaget J (1950) The psychology of intelligence. Routledge and Kegan Paul, London
Prince M (2004) Does active learning work a review of the research. J Eng Educ 93(3):223–231
Raelin J, Coghlan D (2006) Developing managers as learners and researchers: using action learning and action research. J Manage Educ 30(5):670–689
Revans RW (1971) Developing effective managers; a new approach to business education. Praeger Publishers
Sarason Y, Banbury C (2004) Active learning facilitated by using a game-show format or who doesn't want to be a millionaire? J Manag Educ 28(4):509–518
Sutherland TE, Bonwell CC (1996) Using active learning in college classes: a range of options for Faculty in Teaching and Learning in Higher Education. (2008). Acad Manag Learn Educ 7(1):139–142. Retrieved from http://www.jstor.org/stable/40214510
Toptal (no date) Sorting algorithm animations. Retrieved from https://www.toptal.com/developers/sorting-algorithms
Umble M, Umble EJ (2004) Using active learning to transform the monte hall problem into an invaluable classroom exercise. Decis Sci J Innovative Educ 2(2):213–217
University of Huddersfield (2018) About us. Retrieved from https://www.hud.ac.uk/about/
Weltman D (2007) A comparison of traditional and active learning methods: an empirical investigation utilizing a linear mixed model, Ph.D. thesis, The University of Texas at Arlington
YouTube Quick-sort with Hungarian (Küküllőmenti legényes) folk dance. Retrieved from https://www.youtube.com/watch?feature=player_embedded&v=ywWBy6J5gz8