Tareas calificadas por los compañeros: Initiate a conceptual change

Revisar los trabajos de tus compañeros
¡Felicitaciones por enviar tu trabajo! Ahora tus compañeros pueden revisarlo. Para obtener tu calificación, también debes revisar los trabajos de algunos de tus compañeros. Tu calificación debería estar lista antes del **28 de ene. 2:59 -05**.

Revisar tareas

Instrucciones	Mi presentación	Discusiones

Computer science Algorithms course

Enviado el 18 de enero de 2021	Enlace para compartir

CUADRO DE AVISO

1. Choose a science concept you teach. (It is especially good to pick something you know is difficult for students to learn.) Be sure to choose a concept (e.g. adaptation) and not a whole theory (e.g. evolution). Describe the concept here.

Efficiency of algorithms (Time complexity big O notation)

CUADRO DE AVISO

2. Identify typical preconceptions/everyday conceptions on the concept. You can use several resources to identify them (e.g. your experience, colleagues' experience, directly asking your students.) There is also a growing body of publications in educational science journals on preconceptions. This website is a great starting point: http://assessment.aaas.org/topics (click on the most related topic, then click on a subtopic to come to a menu with student misconceptions as an option).

An overview of papers up to 2009 on student conceptions can be found here: http://archiv.ipn.uni-kiel.de/stcse/

Describe here your students' preconceptions and explain how you know about them.

Preconceptions
According to my experience, i think the two main preconceptions on students mind's before learning this topic are:

1. Modern computers, especially corporative devices, are incredibly fast hardware that can run almost any necessary code in an appropiate time.

2. A good way to measure the efficiency of an algorithm is to make experiments measuring the total execution time on a computer using different inputs. This is equivalent to comparing two software programs that do a same task and choosing the best .

CUADRO DE AVISO

3. Compare the science concept and your students' preconceptions to identify your students' learning demands. Describe here what students need to learn to advance their understanding of the science concept.

Concept vs preconceptions
A mathematical tool called Big O asympdotic notation can be used to study an algorithm's efficiency. This is important because some classical and new problems have several theoretical solutions and only the most efficient ones can be actually processed in appropiate time by computers. Empirical approach is not good because as any function, the performance of algorithms varies depending on input, and it is difficult to guarantee enough meditions to obtain global conclussions.

What students are expected to learn

I. When studying a computer program, we can count computational steps or constant time operations.
For example let's see this small example with 7 tasks done:

```
_______________
int a; //create variable 1
a=4; //asign value to variable 2

int b; //create variable 3
b=6; //asign value to variable 4

int c; //create variable 5
c = a+b; //calculate sum (6) and asign result value to variable(7)
_______________
```

II. Because of the fact that we are interested in verifying efficiecy of algorithms and not of code itself and because of the fact that we do care about the overall performance of the algorithm against any input, no matter how big it is, we need to follow some guidelines:

- We'll now follow big O asympdotic notation. which gives a estimation "algorithm makes at most O(n) operations".

- We'll consider an 'n' input where n is a limit to infinity (n Lim-->inf)

-after counting steps done in the algorithm we'll clasify the algorithm according to the highgest term found in the resulting polynomial or expression.

for example let's see the following pseudocode:

```
read n integers and store them in a list;
compare every number in the list one another and sort them. (n^2 comparations done)

show n times a happy face on screen ":)" (n operations done)
```

Because we make n^2 + n operations we can say the algorithm is n^2

Note: while sorting, we had to swap some elements to sort the final list.
The number of swaps is smaller than n^2. so our analysis is still correct.

-We can use constants in the exact expressions of algorithm complexity, but we must remove them when using O notation.

for example his small code:

```
read n
print n times ":D"
print n times ":)"
```

Exact complexity: 1 + n + n = 2n+1
O(n) complexity: n

-In our analysis we assume always the worst case depending on the problem. For example we can assume a inverted list if we are sorting it.

III. The big O analysis is independent of hardware, it is a global mathematical tool for measuring efficiency.

IV. In reality other facts as memory use may be also important to consider.

V. Big O notation is used for asympdotic big values. It's probable that you'll find an algorithm that works really good with small inputs and maybe your daily use of the algorithm is always between the optimal input range, so no problem to choose an algorithm with negative O(n) beheavior. In this sense it's frequent to study graphs of performance time againt input size.

CUADRO DE AVISO

4. Develop an intervention to address the learning demand you identified in question three. Describe the intervention here. It might be helpful to think back to the steps of conceptual change described in the videos this week and K2P briefs 3 and 4.

Intervention

Step 1: Identify preconceptions

A. Computers are fast for most tasks, even long complex structured programs.
B.Empirically testing and measuring the performance of code is a good tool.

Step 2: Cognitive conflict

A. show slow performance of "easy solutions for easy problems" then show optimal solutions performance-time
I would give students the sample code to run themselves and see the interesting results.

Consider the n Queen's problem (you have a chess board and you have to obtain a configuration placing n Queens in such a way that no Queens attack each other). If you test all combinations of choosing n squares in the board and checking that there aren't any conflicts it will be slow to getting the answer. Now use a Backtracking fast solution.

Make a cyclic code to calculate exponentiation x^y. Try big integer values for both x and y. Now use the fast exponentiation algorithm.

B. show empirical vs analytical results of a previously selected algorithm

I would give a code and ask the students to test it with any testcase they like. Then i would show a graphic and the mathematical estimation or efficiency of the given program. It is expected to find signifiicative diferences between empirical and analytical results.

Step 3: scientific concepts
Explanation of asymdotic Big O notation (As explained in question 3).
I would also consider doing these interventions during the class:

-"we are counting and not guessing the exact performance, but even through we then ignore some terms the asymdotic behaviur gives us a very precise and easy for categorization tool" (new topic is plausible)

-we can plot the experimental results vs the analytical result. Shapes will have similarities. We can emphasize higher values beheaviour, to show that even through at some initial points results are different, asyndotycally the analitical model is correct (attractive, new topic is better and easier than previous approximations)

-Show visual Examples using animations and videos that compare algorithms, for example sorting algorithms are vey usual in this sessions (explicative and predictive power).

Step 4: accepting new conceptions
Depends on students

CUADRO DE AVISO

5. Student learning - How would you know if this teaching intervention will lead to an improved good learning outcome for your students?

Before:
-I would ask the opinion of my plan to a proffesor with more experience
-I would write a sketch of my class and compare it to the explanations found in materials like books or webpages and try thinking like an student of my class which was easier to understand.

After:

- I would make a Quiz asking the students to explain what is big O natation and then sorting a group of given algorithms from slowest to fastest. Basically a 2 minute paper + practice exercise.

-I would ask my students to analyse a given algorithm, and then make to randomly choosen students questions like input intervals of fast performance and asymdotic performance. The idea here would be finding unclear points and solving them together.

CUADRO DE AVISO

6. Reflection - How is the teaching you have described in this assignment different from how you usually teach or how you were taught as a student?

When i learned this topic, the approach made by my proffesor was very theoretical and because of that fact it was a bit difficult to connect it to reality and understand its importance. In the described intervention i would emphasize in the practical implications of the explained theory and i would also give a first approach to the use in proffessional algorithm design.

Also, my teacher didn't adressed preconceptions, so understanding the theory was something that took time for me. Actually i learned this topic in the course of data structures, but understood it better in a review class in an algorithms course. In the described plan i made i believe i would make an ordered and clear explanation with analytical and practical examples and also solving problems given due to preconceptions.

Editar envío

Comentarios
Solo el estudiante puede ver comentarios que se dejan para ese estudiante y la persona que dejó el comentario.

JS

Comparte tus ideas...