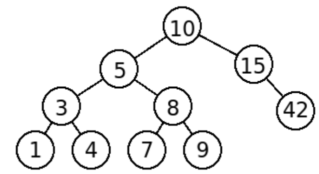
Model Test/Exam (1st Half) - Advanced Topics in Algorithms 2020/2021 (CC4020)

Duration: 1h30m+30m DCC/FCUP

Answer to any 5 out of the next 6 questions (from question 1 to 6) *[17% for each group]*

1 - *Balanced Binary Search Trees.*



a) Suppose you **remove number 42** from the **AVL tree** shown in the right figure. Indicate what would be the resulting tree, justifying your answer.

b) Indicate an **advantage** and a **disadvantage** of **AVL trees** when compared to **red black trees**, justifying why they would be better or worse in each situation.

2 - *Self-Adjusting Data Structures.* Consider you are using a *splay tree*.

a) What does it mean to say that each basic operation (ex: *find, insert, remove*) has an **amortized** complexity of $\mathcal{O}(\log n)$? Can any of these operations have a linear cost? Why?

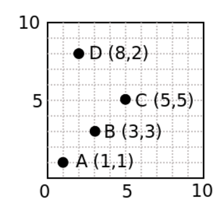
b) What is the purpose of the **splay operation**? Give a brief description of how it works (naming its basic operations) and why it has a fundamental role in the splay trees algorithmic efficiency.

3 - *Probabilistic Data Structures.*

a) Describe an algorithm to **search** for an item in a **skip list** with n items. Can you give an intuitive explanation of why its expected temporal complexity is **logarithmic**?

b) Why doe we say that a **bloom filter** may give **false positives**? Can it give **false negatives**? Why?

4 - *Spatial Data-Structures.*



a) Draw a **point-region (PR) quadtree** storing the 4 points of the right figure. (draw both the tree and the 2D plane representation)

b) Describe a set of points would give origin to a **very unbalanced PR quadtree**? Why? Explain how you could create a **balanced point quadtree** representing the same set of points.

5 - *LCA, RMQ and 1D Data Structures*.

a) Explain how the **LCA** problem can be reduced to the **RMQ** problem. Don't forget to show the associated complexity and exemplify with the same tree as question 1.

b) Give a brief explanation of the concept of **lazy propagation**. In what kind of data structure and operation can it be applied? What would the complexity be with and without its usage?

6 - *String Matching*.

a) Explain the purpose of the π function in the **KMP** algorithm? Show its contents for the pattern $P = abacabab$ and explain what would happen with the first mismatch on searching this pattern in the text $T = abacabacabab$.

b) Draw the **suffix tree** of the string *babanana* and explain how you could find in it the **shortest unique substring**.

Answer to one of the following two questions (question 7 and 8) *[15% for each group]*

7 - Summing subarrays.

Imagine that you are storing n contiguous values (initially set to zero). You want to support two operations: **update(i,v)**, that changes to v the $i-th$ value, and **sum(a,b)**, that returns the sum between positions a and b (inclusive). Indicate a data structure that would allow **sublinear** costs [$o(n)$] for both operations. Explain what would be stored on the data structure, how each operation would be implemented and the associated cost. Illustrate with a simple example.

8 - Searching for students.

Imagine you are storing a set of students information as tuples $(name, age, grade)$. You want to support the operation **search(a_1,a_2,g_1,g_2))** that return the names of the students that have (both) ages in the interval $[a_1, a_2]$ and grades in the interval $[g_1, g_2]$. Indicate a data structure that would allow **sublinear** costs [$o(n)$] for this operation. Explain what would be stored on the data structure, how the operation would be implemented and the associated cost. Illustrate with a simple example.