Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

prof. Fabrizio Ferrandi

Advanced Algorithms and Parallel Programming 1st mid-term test– November 8th 2022

Polimi ID _____ _____

Surname _____ _____ **Name** _____

- This is a closed-book examination. You cannot use computers, phones or laptops during the exam.
- Paper will be provided, but you should bring and use writing instruments that yield marks dark enough to be read easily. Erasable pens can be used.
- Total available time: 1h:30m.

Exercise 1 (4 points) _____

Exercise 2 (4 points) _____

Exercise 3 (4 points) _____

Exercise 4 (4 points) _____

Exercise n. 1

Answer the following questions about parallel application modeling and briefly explain (without an explanation, the answer will be considered invalid)

A. Describe the main characteristics of a PRAM model. (2)

B. Define the speedup when a sequential code is accelerated with p processors. (1)

C. Please describe the differences between the Amdahl and the Gustafson laws. (1)

Exercise n. 2

Answer the following questions about dynamic programming algorithm and briefly explain (without an explanation, the answer will be considered invalid)

A. Describe the Longest Common Subsequence (LCS) algorithm. (2)

B. What kind of temporal or spatial complexity has LCS? (1)

C. LCS is classified as a dynamic programming algorithm. Why? (1)

Exercise n. 3

Answer the following questions about advanced algorithm and briefly explain (without an explanation, the answer will be considered invalid)

A. Please discuss the difference between Online and Offline algorithms. (2)

B. Describe how Treap split/union works.? (2)

Exercise n. 4

Consider a k-bit binary counter counting upwards from 0 using the INCREMENT procedure, implemented according to the following pseudocode:

```
INCREMENT(A, k)
1 i = 0
2 while i < k and A[i] == 1
3   A[i] = 0
4   i = i + 1
5 if i < k
6   A[i] = 1
```

```
RESET(A,k)
  int i = 0;
   while(i < k)
      A[i]=0
   end
```

Provide a pseudocode implementation for the RESET procedure that returns the counter to its original state (all bits set to 0).
Compute the amortized cost for both procedures for a sequence of n operations starting with n-1 INCREMENTs and ending with one RESET using one of the amortized analysis methods seen at lecture.
You are allowed to modify the increment procedure and add additional variables to improve the amortized cost of the RESET procedure.