CS6033 – Design and Analysis of Algorithms I

Homework 1

Manuel — UM-JI (Spring 2021)

Reminders

- Write in a neat and legible handwriting or use LaTeX
- Clearly explain the reasoning process
- Write in a complete style (subject, verb, and object)
- Be critical on your results

*Questions preceded by a * are optional. Although they can be skipped without any deduction, it is important to know and understand the results they contain.*

Ex. 1 — *Hash tables*

In this exercise we want to estimate the maximum number of keys per slot we can expect when inserting n keys into n slots of a hash table.

Given a hash table with n slots, n keys are equiprobably hashed to each slot. Let M denote the maximum number of keys in a slot once they have all been inserted.

1. For any positive integer k, show that the probability P_k that exactly k keys hash to a same slot is

$$\left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k}.$$

2. Prove that the probability P'_k, for the slot with the most keys to have exactly k keys, is less or equal to nP_k.
3. Prove that $P_k < e^k/k^k$.

* 4. Show that for any positive integer $k \geq c \log n/ \log\log n$, for some constant $c > 1$, $P'_k < 1/n^2$.

5. Denoting the expected value of M by $E(M)$, observe that

$$E(M) \leq \Pr\left(M > \frac{c \log n}{\log\log n}\right) n + \Pr\left(M \leq \frac{c\log n}{\log\log n}\right) \frac{c \log n}{\log\log n},$$

and conclude that $E(M) = \mathcal{O}\left(\frac{\log n}{\log\log n}\right)$.

Hint: for question 3 apply Stirling formula.

Ex. 2 — *Minimum spanning tree*

Let G be a graph and T be a minimum spanning tree for G. Write the pseudocode of an algorithm which determines the minimum spanning tree of the graph G when the weight of an edge not in T is decreased.

Ex. 3 — *Simple algorithms*

* 1. Given two n-bits integers stored in two arrays, explain how to compute their sum in an $n+1$-bits array. Write the corresponding pseudocode.

2. One decides to multiply two integers x and y by writing a function `mult(x,y)` returning 0 if one of them is 0 and otherwise returning the sum of $x \cdot (y \bmod 2)$ and a recursive call on `mult`, with parameters $2x$ and $\lfloor y/2 \rfloor$.

a) Express this algorithm as pseudocode.

b) Prove the correctness of this algorithm.

Ex. 4 — *Critical thinking*

1. The *Knapsack problem* is defined as follows. Given a set S and a number n find a subset of S whose elements add up exactly to n. Which of the following algorithms solve the Knapsack problem?
   - Fit the knapsack with the smallest items first.
   - Fit the knapsack with the largest items first.

* 2. In the course (Example (1.26|1.44)) it is mentioned that m should be "a prime not too close from a power of 2" in order for the hash function $H(k) = k \bmod m$ to be a good choice. Explain.

3. Provide an example of a greedy algorithm which is locally optimal while not being globally optimal. Provide all the necessary details to support your claim.

4. Given twenty five horses determine the three fastest ones, in the right order, knowing that no more than five can race at a time. What is the minimum number of races necessary? Detail a general algorithm which solves the problem.

Ex. 5 — *Recursive function*

For any integer n we define its successor $S(n) = n+1$. For any $x, y \in \mathbb{N}$, let f be the function defined by

$$\begin{aligned} f(x,0) &= 0 \\ f(x, S(y)) &= g(f(x,y), x), \end{aligned}$$

where g is such that

$$\begin{aligned} g(x,0) &= x \\ g(x, S(y)) &= S(g(x,y)). \end{aligned}$$

1. Manually calculate $f(0,0)$, $f(1,0)$, $f(1,1)$, $f(2,1)$, $f(2,2)$, and $f(2,3)$.
2. Using words explain what you guess the function f is doing, then write the pseudocode of a clear algorithm to compute f.
3. Test your hypothesis on larger inputs. Either do it manually or run a simple implementation in the language of your choice. by implementing it and testing it on larger inputs.