Advanced information, computation, communication I
EPFL - Fall semester 2020-2021

Week 6

October 20, 2020

Exercise 1.

```
function f1() {
  x=0
  i=1
  while (i ≤ n) {
    x=x+1
    i=x+x
  }
  a=x
}
```

```
function f2() {
  y=0
  j=1
  while (j ≤ n) {
    y=y+1
    j=y*y
  }
  b=y
}
```

After execution of the two program fragments `f1` and `f2`, it is the case that

- ◯ $a \approx \frac{n}{2}$, $b \approx \sqrt{n}$.
- ◯ $a \approx n$, $b \approx \log_2(n)$.
- ◯ $a \approx \frac{n}{2}$, $b \approx \log_2(n)$.
- ◯ $a \approx n$, $b \approx \sqrt{n}$.

Exercise 2. The following three algorithms sort the input sequence $a_0, a_1, \ldots, a_n$ of real numbers in ascending order.

Algorithm 1 Bubble Sort

repeat
  swapped $\leftarrow$ FALSE
  for $i = 1$ **to** $n-1$ **do**
    if $a_{i-1} > a_i$ **then**
      swap a_{i-1} and a_i
      swapped $\leftarrow$ TRUE
    end if
  end for
until swapped = FALSE

Algorithm 2 Selection Sort

for $i = 0$ **to** $n-1$ **do**
  min $\leftarrow i+1$
  for $j = i+1$ **to** n **do**
    if $a_{\min} > a_j$ **then**
      min $\leftarrow j$
    end if
  end for
  if $a_i > a_{\min}$ **then**
    swap a_i and $a_{\min}$
  end if
end for

Algorithm 3 Insertion Sort

for $j = 1$ **to** n **do**
  $i \leftarrow 0$
  while $a_j > a_i$ **do**
    $i \leftarrow i+1$
  end while
  $m \leftarrow a_j$
  for $k = 0$ **to** $j-i-1$ **do**
    $a_{j-k} \leftarrow a_{j-k-1}$
  end for
  $a_i \leftarrow m$
end for

1. Use Bubble Sort, Selection Sort and Insertion Sort to sort the following sequence:

   `9, 12,-43, 20, -2, 3, 7, 28, 19.`

2. How many comparisons are done in each of the algorithms?
3. How many swaps are done in each of the algorithms?
4. What is the approximate overall cost of the two algorithms for an input sequence of length $n+1$?

Exercise 3. Charlotte, Giulia and Patrick are starting university next year. They have applied to EPFL, ETHZ and USI, and their preferences are listed as follows

Student	Most preferable	$\longrightarrow$	least preferable
Charlotte	USI	ETHZ	EPFL
Giulia	EPFL	USI	ETHZ
Patrick	ETHZ	EPFL	USI

The universities, on the other hand, have their own lists of preferred students

University	Most preferable	$\longrightarrow$	least preferable
EPFL	Giulia	Charlotte	Patrick
ETHZ	Giulia	Patrick	Charlotte
USI	Patrick	Charlotte	Giulia

In how many ways can we match students with universities, so that there is no pair of a student and a university who both prefer each other more than the university/student they have been matched up with?

Exercise 4. The stable maximum matching problem is a generalisation of the previous exercise in the following way: we are given two equally sized sets A and B and to each element of the sets is assigned an ordering of preferences for the elements of the other set. A matching between two sets is a pairing between the elements of the sets, i.e., a bijective function from A to B. A matching is called unstable if there exist two elements $a \in A$, $b \in B$ that are not paired and both of them prefer each other to the element they have been paired with. A matching that is not unstable is called stable.

Is a stable maximum matching unique? Either prove that every stable maximum matching is unique, or disprove it with a counterexample.

Exercise 5. Let $\{A, B, C, D\}$ be a set of men, and $\{a, b, c, d\}$ a set of women. We want to match up men and women using the Gale-Shapley algorithm in two different ways. The preferences of men and women are given in the following lists, going from most preferable on the left to least preferable on the right.

Men	1st	2nd	3rd	4th
A	c	d	b	a
B	d	c	a	b
C	a	c	b	d
D	b	d	a	c

Women	1st	2nd	3rd	4th
a	D	A	B	C
b	C	B	A	D
c	C	B	A	D
d	D	A	B	C

1. If the men propose, and women accept/reject, what is the matching after the algorithm terminates?
2. If the women propose, and men accept/reject, what is the matching after the algorithm terminates?
3. Who is the best possible (stable) valid partner for “a”?

Exercise 6.

$$L_{x_1} = (y_3, y_1, y_2) \qquad L_{y_1} = (x_2, x_1, x_3)$$
$$L_{x_2} = (y_2, y_3, y_1) \qquad L_{y_2} = (x_1, x_3, x_2)$$
$$L_{x_3} = (y_1, y_2, y_3) \qquad L_{y_3} = (x_3, x_2, x_1)$$

(*français*) Soit L_x pour $x \in X = \{x_1, x_2, x_3\}$ la liste de préférence de x donnée ci-dessus
et soit L_y pour $y \in Y = \{y_1, y_2, y_3\}$ la liste de préférence de y donnée ci-dessus.
Le couplage $\{(x_1, y_1), (x_2, y_3), (x_3, y_2)\}$ est

(*English*) Let L_x for $x \in X = \{x_1, x_2, x_3\}$ be the preference list of x as given above
and let L_y for $y \in Y = \{y_1, y_2, y_3\}$ be the preference list of y as given above.
The matching $\{(x_1, y_1), (x_2, y_3), (x_3, y_2)\}$ is

○ $\begin{cases} \text{instable.} \\ \text{unstable.} \end{cases}$

○ $\begin{cases} \text{stable et optimal pour } Y. \\ \text{stable and } Y\text{-optimal.} \end{cases}$

○ $\begin{cases} \text{stable et optimal pour } X. \\ \text{stable and } X\text{-optimal.} \end{cases}$

○ $\begin{cases} \text{stable, mais n'est pas un couplage stable optimal pour } X \text{ ou pour } Y. \\ \text{stable but not a stable matching that is } X\text{- or } Y\text{-optimal.} \end{cases}$

Exercise 7. Use the cashier's algorithm to make change using quarters, dimes, nickels, and pennies for:

1. 87 cents.
2. 49 cents.
3. 99 cents.
4. 33 cents.

Exercise 8. Describe an algorithm that determines whether a function from a finite set to another finite set is one-to-one.

Exercise 9. Adapt the bubble sort algorithm so that it stops when no interchanges are required. Express this more efficient version of the algorithm in pseudocode.

Exercise 10. Two strings are anagrams if each can be formed from the other string by rearranging its characters. Devise an algorithm to determine whether two strings are anagrams