CSE530: Algorithms & Complexity
Exercise Set 2

Antoine Vigneron

March 25, 2018

This is not an assignment. These exercises will not be graded. Some of the exercises are taken from the textbook *Introduction to Algorithms* by Cormen, Leiserson, Rivest and Stein.

1. For each of the relations below, write whether it is TRUE (T) or FALSE (F).

- $n^2 = O(n^2)$
- $n^2 = o(n^2)$
- $n^2 = \Omega(n^2)$
- $n^2 = \Theta(n^2)$
- $5 = O(1)$
- $5 = O(0)$
- $(-1)^n = O(1)$
- $n^3 + 2n + 4 = O(n^4)$
- $n^3 = O(n^3 - n + 2)$
- $2^{n+\log n} = O(2^n)$

2. Is the statement below true? Justify your answer.

- "If f is an increasing function and $f(0) = 0$, then $f(2n) = O(f(n))$"

3. Consider the Algorithm 1. Suppose that one execution of each line 2, 3, 4, 5 and 6 takes

Algorithm 1

```
1: procedure DUPLICATE(A[1...n])
2:     for i ← 1, n − 1 do
3:         for j ← i + 1, n do
4:             if A[i] = A[j] then
5:                 return (i, j)
6:     return NOT FOUND
```

constant time c_2, c_3, c_4, c_5 and c_6, respectively. Give the running time $T(n)$ of Algorithm 1 as a function of n and these five constants, assuming that it returns NOT FOUND. Then express this running time using the Θ notation.

4. You are given k types of coins with values $v_1, \ldots, v_k \in \mathbb{N}$ and a cost $C \in \mathbb{N}$. You may assume $v_1 = 1$ so that it is always possible to make any cost. You want to find the smallest number of coins required to sum to C exactly. For example, assume you have coins of values 1, 5, and 10. Then the smallest number of coins to make 26 is 4: take 2 coins of value 10, 1 coin of value 5, and 1 coin of value 1.

Give an algorithm for this problem. Its running time should be polynomial in $n = \max(k, C)$.

5. Using the definitions of flows and flow networks on Slides 5 and 6, Lecture 5, show the following: If $(u, v) \notin E$ and $(v, u) \notin E$, then $f(u, v) = 0$.

6. Let f be a flow network $G(V, E)$ and let α be a real number. The *scalar flow product* αf is a function from $V \times V$ to $\mathbb{R}$ defined by

$$(\alpha f)(u, v) = \alpha \cdot f(u, v).$$

Prove that the flows in a network form a *convex set*. That is, show that, if f_1 and f_2 are flows, then $\alpha f_1 + (1 - \alpha) f_2$ is a flow for any $0 \leqslant \alpha \leqslant 1$.

7. Figure 1 shows a flow network G on which a flow f has been computed. The notation is the same as in Lecture 5: for instance, the label 5/10 on edge (s, v_2) means that $f(s, v_2) = 5$ and $c(s, v_2) = 10$.

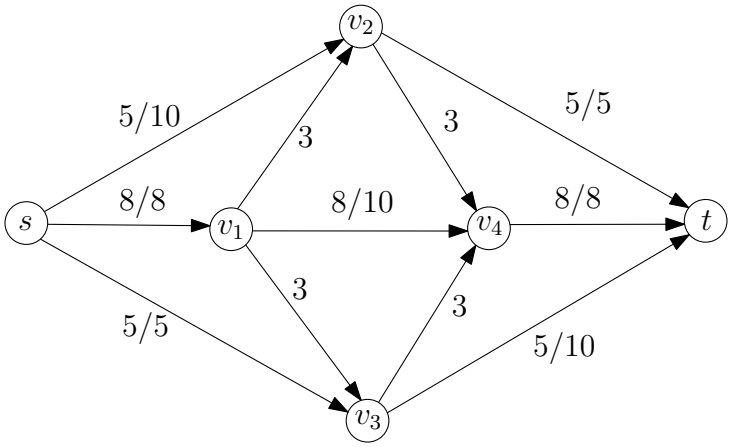

Figure 1: The flow network G and the flow f from Question 7.

(a) Is f a maximum flow? Justify your answer.

(b) Find a minimum cut in G. Justify your answer.

8. Find a minimum $s - t$ cut in the graph drawn in Figure 2. Justify your answer.

9. Consider a set of mobile computing clients in a certain town who each need to be connected to one of several possible *base stations*. We will suppose that there are n clients, with the position of each client specified by its (x, y) coordinates. There are also k base stations; the position of each of these is specified by (x, y) coordinates as well.

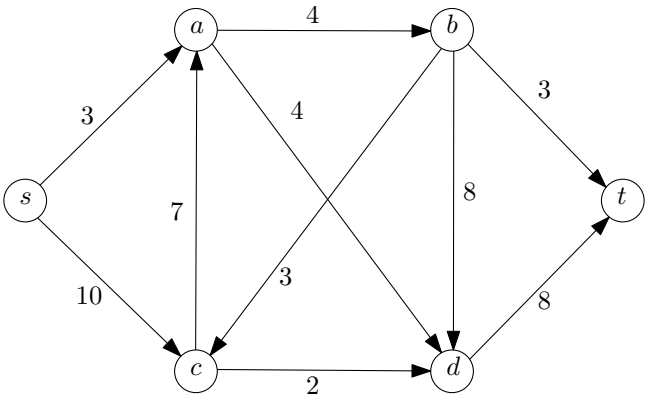

Figure 2: Flow network from Question 8.

For each client, we wish to connect it to exactly one of the base stations. Our choice of connections is constrained in the following ways. There is a *range parameter* r–a client can only be connected to a base station that is within distance r. There is also a *load parameter* L—no more than L clients can be connected to a single base station.

Your goal is to design an efficient algorithm for the following problem. Given the positions of a set of clients and a set of base stations, as well as the range and load parameters, decide whether all clients can be connected simultaneously to a base station, subject to the range and load conditions in the previous paragraph.

Give a polynomial-time algorithm for this problem and prove that it returns a correct answer.

11. We consider the *image segmentation* problem: Given a digital image, we want to partition its pixels into two classes. For instance, we may want to separate an object from the background.

The problem is formalized as follows. The image is a set of pixels $E = \{1, \ldots, n\}$, to be partitioned into two classes A and B. For each pixel i, we are given the likelihoods $a_i > 0$ and $b_i > 0$ that it belongs to class A and B, respectively. In order to minimize the size of the boundary between A and B, we are also given a penalty $p_{ij} > 0$ for each pair of pixels: We need to pay the penalty $p_{ij} = p_{ji}$ if i and j belong to different classes. The goal is to maximize the sum of the likelihoods, minus the penalties:

$$\text{Maximize} \quad q(A, B) = \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in A \times B} p_{ij} \quad \text{over all partitions } \{A, B\} \text{ of } E.$$

Give a polynomial-time algorithm for this problem.