Sabancı University
Faculty of Engineering and Natural Sciences
CS301 – Algorithms
Final Exam

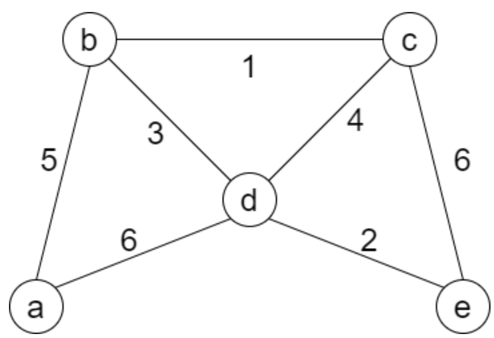
January 22, 2021 @ 10:45–11:30

PLEASE NOTE:

- Provide only the requested information and nothing more.
- Unreadable, unintelligible and irrelevant answers will not be considered.

Question	Maximum Points	Points
1	15	
2	20	
3	10	
4	15	
Total	60	

Question 1) [15 points] Find a minimum spanning tree for the following graph:



(a) (10 points) by using Prim's algorithm (take d as the starting node and fill in the following table).

Iteration No	Covered Nodes	Light Edge
1	$\{d\}$	(d, e)
2	$\{d, e\}$	$\ldots$
3	$\vdots$	$\vdots$

(b) (5 points) by using Kruskal's algorithm (fill in the following table)

Iteration No	Selected Edge
1	(b, c)
2	$\ldots$
3	$\vdots$

Question 2) [20 points]

(a) (10 points) Draw the sorting network for 5 inputs that use the idea of the selection sort.

(b) (5 points) Consider a sorting network design by using the idea of the selection sort as in part (a), but this time imagine the network designed for n input lines. Write down the recurrence for the number of comparators in this sorting network:

$$C(n) = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

(c) (5 points) Give an asymptotic **tight bound** for the solution of the recurrence in part (b). You can simply state the bound, you don't need to show your steps.

Question 3) [10 points] In a flow network, let's call an edge "*a blocking edge*" if increasing only the capacity of this edge increases the maximum flow value.

Does every network have a blocking edge? If yes, explain why, if not give an example of such a network?

Question 4) [15 points] Consider an algorithm A, which returns the correct result with the probability $1/n$ for a problem of size n. Suppose that we have a verifier algorithm V which we can use to check if the result returned by algorithm A is correct or not. Suppose that we develop the following algorithm to apply the amplification technique, where we use A to generate a candidate solution and use V to verify it repeatedly:

```
A() {
// find and return a candidate solution x
...
}

V(x) {
// check if x is a correct solution
...
}

Amplified_A(c) {
  for i = 1 to M do
    call A to find a candidate solution x
    call V to check if x is a correct solution
  end for
}
```

(a) (10 points) What should be the value of M used in the algorithm Amplified_A so that the probability of failing to find the correct answer for Amplified_A is at most

$$p = \sqrt{((n-1)/n)^c}$$

Show the steps of your calculation.

(b) (5 points) Would you consider algorithm A as Monte Carlo or Las Vegas? Why?