

**Teoritenta i Algoritmer (datastrukturer) och komplexitet
för KTH DD2350–2352 2019-05-31, klockan 8.00–11.00**

No aids are allowed. 12 points are required for grade E, 15 points for grade D and 18 points for grade C. If you have done the labs you can get up to 4 bonus points. If you have got bonus points, please indicate it in your solutions.

If you are registered on DD2350, 13 points are required for grade E. Bonus points from DD2350 are counted in the exam.

In all problems you can assume $P \neq NP$.

1. (8 p)

Are these statements true or false? For each sub-task a correct answer gives 1 point and an answer with convincing justification gives 2 points.

- (a) The problem of deciding if an undirected graph is bipartite is NP-Complete.
- (b) If a Divide and Conquer-algorithm has a time complexity $T(n)$ given by the recursion formula

$$T(n) = 3T\left(\frac{n}{3}\right) + cn$$

then $T(n) \in O(n^3)$.

- (c) It is possible to construct a Turing Machine that decides if there are negative cycles in a directed graph.
- (d) All problems that are in NP are also in P.

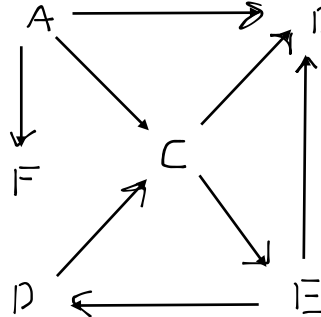
2. (3 p)

Let us assume that we have a directed graph G and we use the Ford-Fulkerson algorithm to find maximal flows between s and t in G .

- (a) If all capacities for the edges are integers, then the maximum flow must have an integer value. Carefully explain why.
- (b) We say that two directed paths are *edge disjoint* if there is no edge belonging to both paths. Let us assume that there is a set of m pairwise edge disjoint paths between s and t and m is maximal in this sense. Let us also assume that all edges have capacity 1. Then the maximum flow must have value m . Carefully explain why.

3. (3 p)

In the graph below the nodes are problems. An arrow like $A \rightarrow B$ indicates that there is a polynomial time reduction from A to B . Observe that there could be more reductions than the ones indicated.



Let us assume that C is NP-Complete. Answer these questions:

- Which problems must be NP-Complete?
- Which problems could be outside NP?
- Given $P \neq NP$, which problems could then be in P?

4. (3 p)

We can define a problem NEF (Not Equivalent Formulas) that takes two propositional logic formulas F_1 and F_2 as input. The goal is to decide if the formulas are not equivalent. So a yes answer means that the formulas are *not* equivalent.

- Show that this problem is in NP.
- Show how you can reduce SAT to this problem. More specific, given an instance ϕ to SAT, show how you can choose two formulas F_1, F_2 such that ϕ is satisfiable if and only if F_1 and F_2 are not equivalent.

Extra question for the sophisticated: What is the reason for choosing "not equivalent" instead of the perhaps more natural "equivalent". You don't have to answer this question and answering gives no points but it could be good to think about this.

5. (3p)

- In the course book and in the lecture notes there is a description of an *approximation algorithm* for the problem VERTEX COVER. Describe the algorithm. What is the approximation quotient?
- Instead of using this algorithm we might want to use another simple greedy algorithm: Given G , choose a vertex v of maximal degree. Put v in the vertex cover. Remove v and all edges on v . This gives us a new graph G' . Use the previous step recursively to get a vertex cover.
Give an example when this simple algorithm does not give an optimal size vertex cover.

- (c) A K -regular graph is a graph where each vertex has degree K . Let us assume that we have K -regular graphs with K *fixed*. Show that in this case the algorithm described in b. approximates VERTEX COVER within a factor B . Find a value for B as a function of K .