Nonassignment 3

Advanced Computing Concepts for Engineering

2017. Do not hand in.

Q0 Consider the following problem.

- Input: An array of numbers a, of length N, and another array of numbers b, also of length N.
- Output: An array p of length N of integers such that, for each $i \in \{0, ..N\}$, $p'(i)$ is the number of numbers in array a that are less than $b(i)$.
- Assume that the only operations allowed on arrays a and b are comparing items using $<$ and swapping items of the arrays.
- Show that $\Omega(N \log N)$ is a lower bound for this problem.

Hint. Use proof by contradiction. Proceed by showing that, if the complexity of this problem were not in $\Omega(N \log N)$, then the complexity of comparison-based sorting would also not be in $\Omega(N \log N)$.

Q1 Find an efficient (linear time) algorithm to apply a permutation p to an array a of length n. In particular, suppose that p is a permutation (i.e. a one-one total function from $\{0, ..n\}$ to $\{0, ..n\}$) and that the sequence $[a(p(0)), a(p(1)), ..a(p(n))]$ is in sorted order. Find a way to rearrange the items of array a so that a' is sorted. Your method should take linear time and only constant space in addition to a and p. [Hint: pick a good invariant.]

Q2 A propositional formula is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of atoms. For example

$$(a \wedge \neg b \wedge c) \vee (a \wedge \neg b \wedge \neg c)$$

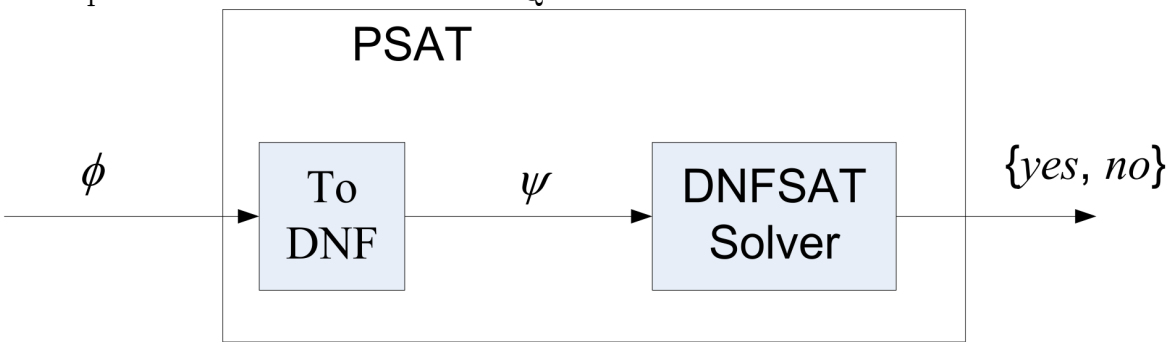
is in disjunctive normal form.

(a) Outline a polynomial time algorithm to determine whether a disjunctive normal form formula is satisfiable.

(b) Outline an algorithm to convert any formula ϕ to an equivalent formula in DNF.

(c) Explain all flaws and gaps in the following "proof" that $\mathbf{P} = \mathbf{NP}$.

We can solve PSAT as follows: Take the input formula ϕ and convert it to an equivalent DNF formula ψ, using the algorithm from part (b). Then use the algorithm from part (a) to determine whether ψ is satisfiable. Output that answer. Since ϕ is satisfiable iff ψ is satisfiable and since the satisfiability of ψ takes only polynomial time to determine, we have a polynomial time algorithm for PSAT. Since PSAT is in **NPC**, the existence of a polynomial time algorithm for it implies **P** = **NP**. Thus **P** = **NP**. Q.E.D.



Q3. Consider the following problem.

The cloth cutting problem.

Input: a set of nonoverlaping rectangles representing one or more bolts of cloth and a set of polygons representing pieces of cloth to be sewn into one of more garments. All coordinates defining the rectangles and polygons are integers.

Output: The answer to the following question: can the set of polygons be arranged so that they all fit within the rectangles without any overlaps and such that each polygon contained within one rectangle. The polygons may only be translated. They may not be rotated, flipped, or resized in any way.

Size: The size of each input is the number of vertices needed to represent the rectangles and polygons.

(a) Explain why this problem is in NP.

(b) Show that this problem is NP-complete.

Hint for part (b): Look up the partition problem (e.g. on Wikipedia) and show that there is a reduction from the cloth cutting problem to the partition problem.

Q4. In the class notes the is an efficient algorithm for finding a string pattern in a long target string. A similar algorithm is discussed in Theories of Computation. Suppose instead of string, the pattern is given by a regular expression. Find an algorithm to determine the first location of a string described by the regular expression x in a string t in time that is linear with respect to the length of t. That is, your algorithm should output the smallest number i such that there is a j such that $t[i, ..j] \in L(x)$, or it should output -1 to indicate

that there is no such number. [Hint: Consider a DFR to be a generalization of a string pattern.]