CSE 101 Homework 3

Due November 18, 2004

1. Let \((a_1, \ldots, a_n)\) be a sequence of \(n\) natural numbers. An increasing subsequence is a subsequence \((a_{i_1}, a_{i_2}, \ldots, a_{i_k})\) such that \(1 \leq i_1 < i_2 < \cdots < i_k \leq n\) and \(a_{i_1} \leq a_{i_2} \leq \cdots \leq a_{i_k}\).

(a) Describe an exhaustive search algorithm for finding the longest increasing subsequence. What is its running time?

(b) Describe a backtracking algorithm that runs significantly faster. Explain the idea behind the algorithm.

(c) What is the worst-case running time of the algorithm from (b) (in \(O\)-notation)? Explain.

(d) Can you come up with an input that achieves the worst-case running time from (c)?

2. Consider the following "greedy" algorithm for finding an independent set in a graph:

\begin{verbatim}
1  S <- { all nodes of G }
2  P <- { }
3  while S not empty
4     s <- node in S with smallest degree
5     add(P, s)
6  S <- S - s - adj(s)
\end{verbatim}

(a) Prove that this algorithm always finds a maximal independent set.

(b) Prove that this algorithm does not always find a maximum independent set.

3. Due to a bad economy, upon graduating from UCSD you become a burglar. One night you are robbing someone’s house. You have a knapsack that can fit \(d\) pounds of loot before your spine telescopes. The house has \(n\) items with weights \(c_1, c_2, \ldots, c_n\). You want to check whether there is some subset of items that will exactly fill up your knapsack. The problem is that the cops are on their way and they’ll be there in time \(2^{3n/4}\), so you have to hurry up. You need to come up with an algorithm that runs in time \(poly(n)2^{n/2}\), where \(poly(n)\) is some polynomial in \(n\).

The algorithm will be not quite divide-and-conquer, but not quite backtracking; you could call it divide-and-backtrack. The algorithm should have the form:

(1) Divide the items into two groups of size \(n/2\)
(2) Use a backtracking algorithm on each group
(3) Find the answer given the two outputs from part (2)

Describe how to do each step and how long the step takes. For step (2), make sure you say what the output of the backtracking algorithm is.