CS524
HW#3

DUE: Wednesday, October 19

1. (4 points) Consider the following problem:
   INPUT Array A of n integers
   OUTPUT $1 \le i, j \le n$ such that for all $1 \le k, l \le n$, $A[i] - A[j] \ge A[k] - A[l]$

a Show an $O(n)$ upper bound on the complexity of this problem.
b Show an $\Omega(n)$ lower bound on the complexity of this problem.

2. (3 points) Assume you want to break into an account by guessing a user's password from among n possible passwords. There is an oracle (a login program) which tells you if you have guessed the proper password. What is the expected number of passwords you must try if you suffer from amnesia (that is, you sample passwords with replacement or you always forget which passwords you have tried)?

3. (15 points) In the MINIMUM SET COVER PROBLEM, we are given a set of sets, $\{S_1, ..., S_n\}$, and we seek a minimum cardinality, k, subset of these sets $S_{j_1}, ..., S_{j_k}$ such that $\bigcup_{1 \le i \le m} S_i = \bigcup_{1 \le i \le k} S_{j_i}$. This is a very difficult problem. In fact, finding a minimum set cover is NP-Hard and testing if there exists set cover of cardinality k is NP-Complete. We are going to study the following Greedy Algorithm, which iteratively adds to the cover a set with a maximum number of uncovered elements until every element is covered.

GREED($\{S_1, ..., S_n\}$)
  COVER $\leftarrow \varnothing$
  UNCOVERED $\leftarrow \bigcup_{1 \le i \le n} S_i$
  UNUSED $\leftarrow \{S_1, ..., S_n\}$
  while UNCOVERED $\ne \varnothing$ **do**
    Let S^* be a member of UNUSED of maximum cardinality
    for each $S \in$ UNUSED **do** $S \leftarrow S - S^*$
    UNCOVERED $\leftarrow$ UNCOVERED-S^*
    UNUSED $\leftarrow$ UNUSED $- \{S^*\}$
    COVER $\leftarrow$ COVER $\cup \{S^*\}$
  return COVER

a Describe an instance of the problem for which GREED fails to find a minimum cover.

For the rest of this problem, we want to find a bound on how poorly GREED can do. We define COVEROPT to be an optimum cover and we define τ =|COVEROPT|. We now bound how well the first τ sets added to COVER approximate COVEROPT.

b What fraction of $\bigcup_{1 \le i \le m} S_i$ (as a function of τ) do we know will be covered by the first set added to COVER?

c What fraction of $\bigcup_{1 \le i \le m} S_i$ (as a function of τ) do we know will be covered by the first τ sets added to COVER? Hint: $1/e \ge \left(1-1/\tau\right)^{\tau}$

C.S.524

SOLUTION FOR H.W. #3

1. $A[i]$ will be a maximum element of A, and $A[j]$ will be a minimum.

a The following algorithm provides a $O(n)$ upper bound.

$MAX \leftarrow -\infty$

$MIN \leftarrow \infty$

for $k \leftarrow 1$ **to** n **do**

  if $A[k] > MAX$ **then** { $i \leftarrow k$

    $MAX \leftarrow A[k]$ }

  if $A[k] < MIN$ **then** { $j \leftarrow k$

    $MIN \leftarrow A[k]$ }

return i,j

b Suppose an algorithm solves the problem in less than linear time. Then some element, say $A[k]$, wasn't checked. If the algorithm returns $i=k$, then change $A[k]$ to be $-\infty$. Otherwise, change $A[k]$ to be ∞. Leave all other elements of A unchanged. The algorithm gets exactly the same answers to its queries on the new values of A as on the old values. But now it gives the wrong answer. This contradiction shows that any algorithm to solve the problem must take linear time.

2. The probability of success is $1/n$, and each trial is independent of all other trials due to amnesia. Letting rv X denote the number of trials, which can be any positive integer as long as $n>1$, it follows that the probability of needing $k \geq 1$ trials is the probability of k-1 failures followed by 1 success, that is $\left(\frac{n-1}{n}\right)^{k-1}\frac{1}{n}$. Thus $E[X] = \sum_{k\geq 1} k\left(\frac{n-1}{n}\right)^{k-1}\frac{1}{n} = n$, where the last equality is established in the text and in class.

3. ***a*** A minimal cover of $\{\{1,2,3,4\},\{1,2,6\},\{3,4,5\}\}$ of cardinality 2 is

COVEROPT=$\{\{1,2,6\},\{3,4,5\}\}$, but GREED constructs

COVER=$\{\{1,2,3,4\},\{1,2,6\},\{3,4,5\}\}$ of cardinality 3.

b Some set in COVEROPT must contain $\bigcup_{1\leq i\leq m} S_i/\tau$ elements, so there must be a set in $\{S_1,...,S_n\}$ with at least this many elements. And GREED will choose a set at least this large.

c After the first execution of the **while**-loop in GREED, the uncovered fraction of $\bigcup_{1 \le i \le m} S_i$ is at most $\frac{\tau - 1}{\tau}$. The next execution of the **while**-loop will cover at least the fraction $\frac{1}{\tau}$ of this fraction, or $\frac{1}{\tau}\frac{\tau - 1}{\tau}$ of $\bigcup_{1 \le i \le m} S_i$, and the uncovered fraction will be at most $\left(\frac{\tau - 1}{\tau}\right)^2$. After τ iterations the uncovered fraction will be at most $\left(\frac{\tau - 1}{\tau}\right)^\tau$, and the covered fraction will be at least $1 - \left(\frac{\tau - 1}{\tau}\right)^\tau \ge 1 - \frac{1}{e}$.