CS 4820, Spring 2017 **Homework 3, Problem 2**
Name: Yuxiang Peng
NetID: yp344
Collaborators:jl3455, zl542

(2) *(10 points)*
Some biologists have asked for your help in designing an experiment that involves analyzing segments of the gene sequence of a chromosome. The chromosome they are studying has n *loci* numbered $1, 2, \ldots, n$, and a designated subset of these loci are deemed *interesting*. To study the interesting loci, the biologists will be cutting out short segments of the DNA using *restriction enzymes*. There are m of these enzymes; each one cuts out a specific segment of the loci, $[a_i, b_i] = \{a_i, a_i{+}1, \ldots, b_i{-}1, b_i\}$. However, the experimental set-up limits the biologists to using only $k < m$ of the restriction enzymes. They want to include as many interesting loci as possible among the segments that will be produced.

We thus have an instance of the following optimization problem. One is given:

- the values of n,m, and k;
- a subset of $\{1, \ldots, n\}$ that lists the interesting loci;
- the endpoints a_i, b_i for each $i = 1, \ldots, m$.

Let us say that *enzyme i covers locus j* if it is the case that $a_i \leq j \leq b_i$. The objective is to choose a k-element subset $R \subset \{1, \ldots, m\}$ to maximize the number of interesting loci that are covered by at least one $i \in R$. Design a polynomial-time algorithm to compute this set R. (Note that your algorithm should output the set of restriction enzymes, R. It does not need to output the number of interesting loci covered by R, nor the locations of those loci.)

Providing that:
1. The enzyme segments are sorted by their last loci in ascending order(m=1,2,...,M).
2. We know $V(a_m, n)$ which is the number of interesting loci between two loci a_m and n. M(m,k,n)=Opt(m,k,n) for all m,k,n.

```
Algorithm:
Initialize M[0,k,n]=0, M[m,0,n]=0, M[m,k,n]=0
For m from 1 to M
For k from 1 to K
For n from 1 to N
if $a_m > N$
M(m,k,n)=M(m-1,k,n)
else
M(m,k,n)=max{M(m-1,k,n), $V(a_m, n)$+M(m-1,k-1,$a_m - 1$)}
Return M(M,K,N)
```

Opt(m,k,n)=The value of optimal solution in which case we can choose at most k enzyme segments from m enzyme segments to maximize the total number or interesting loci from loci 1 to n.

Runtime: $O(M * K * N)$.

Proof
Hypothesis: M(m,k,n)=Opt(m,k,n) for all m,k,n.
Because M(0,0,0)=0 is correct.
If segment m is not in optimal solution: Opt(m,k,n)=Opt(m-1,k,n). If segment m is in the optimal solution,Opt(m,k,n)=$V(a_m, n)$+Opt(m-1,k-1,$a_m - 1$). And there are only these two options for a single segment. Because Opt(m,k,n) is correct, Opt(m+,k+,n+) is correct since it is calculated based on previous recursive functions which are assumed right. The recursion is back to the base status which is correct too.