**CS 4820, Spring 2019** **Homework 5, Problem 1**
Name: Kevin Klaben NetID: kek228 Collaborators Quinn Lui

**(1a)** *(5 points)* Design a dynamic programming algorithm that computes, for every pair $i, j$, the quantity $q_{ij} = \Pr(X_1 + \cdots + X_i = j)$, and then outputs the sum $\sum_{j=a}^{b} q_{kj}$.

**You may omit the proof of correctness, but you should analyze the running time of this dynamic programming algorithm.**

**(1b)** *(10 points)* Design an algorithm that computes $\Pr(a \leq X \leq b)$ in time $O(k\,n \log(k) \log(kn))$.

**For this part of the problem, include both the running time analysis and the proof of correctness.**

*HINT: If $Y$ and $Z$ are independent random variables taking values in $\{0, 1, \ldots, n-1\}$, show that the probability distribution of their sum $Y + Z$ can be computed as the convolution of two vectors representing the probability distributions of $Y$ and of $Z$.*

```
T[k][n+1]=Null
RECPOL(l,m)
if T[l][m]!=NULL then
  return T[l][m]
else
    sum=0
    if (l==k) then return sum=P[k][m]
  else
    for x=m; x>=0;x−− do
      sum=sum+P[l][x]*RECPOL(l+1,m-x)
    end for
  end if
  T[l][m]=sum
  return T[l][m]
end if
```

Runtime Analysis:
This algorithm will be O(k$n^2$) as the dynamic programming table is k by n and for each entry the for loop could iterate up to n times, thus overall running time is O(k$n^2$).

(1b) The Algorithm:

```
T[k][n+1]=Null
RECPOL2(start,end)
if (start==end) then
  return the column of P for $X_start$ with n zeros on the end
else if (start==end-1) then
  return the computed convolution of the column of P for $X_start$ and the column of P for
  $X_end$ using the convolution algorithm stated in class
```

```
else
    mid=compute the integer midpoint between start and end
    return the convolution of RECPOL2(start, mid) and RECPOL2(mid+1,end)
end if
```

Calling RECPOL2(0, k) and then taking the sum from a to b over the entries of the returned vector from this call will yield the desired value.

Runtime Analysis:
The Runtime of this algorithm is O(knlog(k)log(kn)). In analyzing the depth of recursion we can see that by dividing the number of random variable in half each time and the number of variables is k we see that the depth of recursion is log(k). The convolution algorithm from class runs in time nlog(n) where n is the number of elements in the vectors being convolved. As the vectors could have a maximum length of k(n-1) thus we have a convolution algorithm taking knlog(kn) time. Thus, overall the RECPOL2 algorithm will have an overall running time of O(knlog(k)log(kn)). After RECPOL2 is called the returned vector of values is iterated over and summed form a to b. At a maximum this could be kn O(1) operations and thus the overall running time will be O(knlog(k)log(kn))+O(kn)=O(knlog(k)log(kn))

Proof of correctness:
Claim: The convolution of the columns of P will result in the single vector which holds the probabilities that X=the number entry of the column.
Proof: By induction on k WTS k=2 that the claim holds and k=m given k=m/2 hold the correct probabilities.

Case: k=2
Proof: In the case where k is 2 we have only two vectors, a and b. The convolution of two vectors of some arbitrary length n will involved extending both vectors with n additional zeros. Next, the convolution algorithm from class shows that each entry $e_l$ of the resulting vector e, will be the sum of all $a_i$*$b_j$ where $a_i$ is the ith entry of a and $b_j$ is the jth entry of b where i+j=l. As the ith entry of a is the probability that the random variable represented by a will take on the value i and the jth entry of b is the probability that the random variable represented by b will take on the value j. As these random variables are independent the multiplication of these probabilities can be interpreted as the probability that both events occur. As the convolution algorithm will compute the sum of all situations in which i+j=l we can say that the $e_l$ is the probability that $X_a$+$X_b$=l. As this is the desired outcome the claim for k=2 holds.

Case: k=m given the case for k=m/2 contains the correct values.
Proof: We are given that two vectors v and w contain the probabilities that all m/2 random variables they represent will sum to the value of the entry number. Similarly to the case above for k=2, when these vectors are convolved the entries in the resulting vector will represent the probabilities that $X_1$+$X_2$=l where $X_1$=sum of the random variables v represents, $X_2$=sum of the random variables w represents, and l is the entry number in the resulting output vector. As this vector will contain the correct probabilities, the claim holds. Thus, overall the algorithm will rpoperly compute the probabilities of X taking on values i=0..k(n-1) and store them in the ith entry of the output vector. The probability of X taking on values is simply the sum form the ath entry to the bth entry and thus as this is what the algorithm does, the algorithm is valid.