Advanced Algorithms - Mandatory Assignment 2

Frank Andersen [fand@itu.dk]

December 2020

Advanced Algorithms - Problem set 2, resubmission (in blue)

1. Some exercises on probability:

   (a) Say we want to model the random experiment of throwing n unbiased coins. How many elements are contained in the sample space, and how many possible events are there?

   Answer:
   When looking at our sample space, every time we add another coin flip, an extra 2 options are possible at the end of the sequence, this relation means the sample space are of size:

   Sample space: $\Omega = \{H, T\}^n = 2^n$

   e.g.
   n=1: $|\Omega = 2|^1 = 2 \rightarrow (H, T)$
   n=2: $|\Omega = 2|^2 = 4 \rightarrow (HH, HT, TH, TT)$
   n=3: $|\Omega = 2|^3 = 8 \rightarrow (HHH, HHT, HTH, THH, TTT, TTH, THT, HTT)$

   The number of events that could be obtained from this sample space, is the number of possible subsets of the sample space. The number of possible subsets of a set of size x is 2^x, since our sample space size is 2^n, we get the following number of events for this sample space:

   Possible events: $2^{|\Omega|} = (2^2)^n = 2^{2n}$

   (b) Give probability spaces with events A; B such that:
   i. $Pr(A|B) > Pr(A)$
   ii. $Pr(A|B) = Pr(A)$
   iii. $Pr(A|B) < Pr(A)$

   Answer:

   For the first case, we construct a sample space consisting of flipping 4 unbiased coins. The event B is the subset, where the first 2 coins are heads. The event A is the subset where all coins are heads. Below are shown the probabilities:

   $Pr(A) = \frac{1}{2^4} = \frac{1}{16}$ $\qquad Pr(B) = \frac{1}{2^2} = \frac{1}{4}$ $\qquad Pr(A \bigcap B) = \frac{1}{2^4}$

   $$Pr(A|B) = \frac{Pr(A \bigcap B)}{Pr(B)} = \frac{\frac{1}{2^4}}{\frac{1}{2^2}} = \frac{1}{4}$$

   $$Pr(A|B) > Pr(A) \rightarrow \frac{1}{4} > \frac{1}{16} (\checkmark)$$

   For the second case, we construct a sample space consisting of flipping 2 unbiased coins. The event B is the subset where the first coin is a head. The event A is the subset where the second coin is tails. Since these two events are independent, the probability of A conditioned on B, is equal to the probability of A. The probabilities are shown below:

$$Pr(A) = Pr(B) = \frac{1}{2}$$

$$Pr(A|B) = \frac{Pr(A \bigcap B)}{Pr(B)} = \frac{Pr(A)(B)}{Pr(B)} = \frac{\frac{1}{2} \cdot \frac{1}{2}}{\frac{1}{2}} = \frac{\frac{1}{4}}{\frac{1}{2}} = \frac{1}{2}$$

$$Pr(A|B) = Pr(A) \rightarrow \frac{1}{2} = \frac{1}{2}\ (\checkmark)$$

For the third case, we construct a sample space consisting of flipping 2 unbiased coins. The event B is the subset where the first coin is a head. The event A is the subset where both coins are tails. Since event B happening, means that event A cannot happen, the probability of $Pr(A|B) = 0$. The probabilities are shown below:

$$Pr(A) = \frac{1}{2^2} = \frac{1}{4} \qquad Pr(B) = \frac{1}{2} \qquad Pr(A|B) = 0$$

$$Pr(A|B) < Pr(A) \rightarrow 0 < \frac{1}{4}(\checkmark)$$

(c) Let p_n be the probability that a string $x \in \{0,1\}^n$, chosen uniformly at random, does not contain two consecutive 1-entries. Show that $p_n \to 0$ as $n \to \infty$. (Hint: One particular way of showing this would split each string $x \in \{0,1\}^n$ into small pieces, determine the probability that each piece satisfies the desired property, and then use this to derive an upper bound of $O(c^n)$ for $c < 2$ on the number of $x \in \{0,1\}$ satisfying the property.)

2. Consider a random experiment that succeeds with probability at least p. The experiment can be repeated.

(a) Give a lower bound on the probability of succeeding at least once in t independent trials. (Hint: Consider the event that you fail in t trials. You want to avoid this event.)

Answer:
We are looking for the probability that we have at least 1 success in t trials, or in other words, the probability the not all trials fail. Therefore, if we have the probability of all the trails failing, we can subtract it from 1, to obtain the probability that we have at least 1 success in t trials: The probability of failing all t independent trials is as follows, where Pr(x) is the probability of at least x successes:

$$Pr(1) = 1 - ((1-p)^t)$$

Because the experiment succeeds with at least p, this probability acts as a lower bound on succeeding at least once.

(b) Depending on p, determine a number $c \in \mathbb{N}$ such that, with probability at least 0.99, the experiment succeeds at least once in c trials. (Hint: This is just calculation.)

Answer:
Solution (resubmit):

We consider the Binomial(t,P) of the repeated experiment. We denote p as chance of success
So chance of total failure will be $(1-p)^t$ and therefor we get $1-(1-p)^t$ for all being successful.

Number of minimum trials c needed for atleast 99 % chance of success.
$1-(1-p)^c = 0.99 \Leftrightarrow (1-p)^c = 0.01 \Leftrightarrow$
$\log(1-p)^c = \log(0.01) \Leftrightarrow c \cdot \log(1-p) = log(0.01) \Leftrightarrow c = \frac{\log(0.01)}{\log(1-p)}$

(c) In the lecture, we analyzed Karger's algorithm and found that, in a graph with n vertices, it succeeds in finding any fixed minimum cut with probability at least

$$\frac{2}{n(n-1)}$$

How many repetitions of Karger's algorithm are needed to make sure that, with probability at least 0.99, it succeeds at least once? Use that $1-x \leq e^{-x}$ to show that the number of repetitions is $O(n^2)$.

Answer:
Solution (resubmit):

We denote X to be the set of n-bit strings without consecutive 1's.
And Y be the set of n-bit strings where each 2-block is not 11, $2k-1, 2k$ for $k \leq n/2$.
We have a total of $\frac{2}{n}$ blocks, with three possible allowed values 00,01,10.
Legal: [01|10|...]
Not legal: [01|11|...]

$$X \subseteq Y \Rightarrow |X| \leq |Y| = 3^{\frac{n}{2}}$$

So $3^{\frac{n}{2}}$ is equal to $\sqrt{3}^n$ as $\sqrt{3} < 2$ so we have

$$\Rightarrow Pr(x \in X) \leq Pr(x \in Y) = (\frac{\sqrt{3}}{2})^n$$

Because $\frac{\sqrt{3}}{2}$ is less than 1, we have a function that when n goes up in value, the probability descents to 0.

3. You just finished your business trip to Zamazon and would like to collect business cards from its employees. Zamazon is a young start-up and thus extremely serious about portraying itself as fun, so it has meticulously designed an engaging and fun procedure for business card collection: The n employees align in a queue in front of you. The first employee in the queue (counting from the back) passes his business card to the second employee. The second employee either passes the card he received to the third employee (with probability 12) or discards the card and passes his own card to the third employee (with probability $\frac{1}{2}$). In general, the k-th employee either passes on the card he has received (with probability $1-\frac{1}{k}$) or discards the card he received and passes on his own card (with probability $\frac{1}{k}$). You are the recipient of the card passed on by the n-th employee. All of this is synchronized to upbeat dance music.

(a) After one run of this procedure, what is the probability that you receive the business card of employee k for $k \in \{1, \ldots, n\}$? Show how you derived the solution.

Answer:
In order to find the probability that you receive the business card of employee k, out of n employees. To try and solve this, we wrote up the probability tree for the problem, and discovered that a the probability of receiving a card from any employee k is $\frac{1}{n}$. Our approach can be seen in the image below, where P = passing the card to the next employee, and D = Discarding the card and passing your own to the next employee.

We are unsure of how to formally prove that this is the case for any number of employees n.

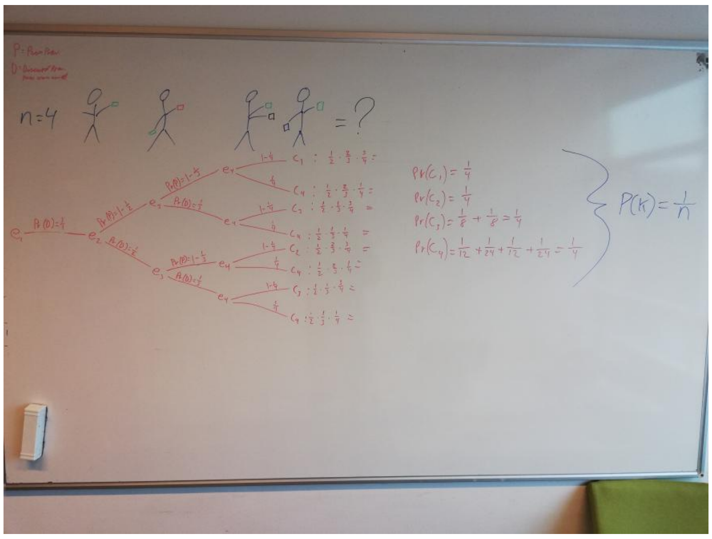
n=4
=?
$Pr(C_1) = \frac{1}{4}$
$Pr(C_2) = \frac{1}{4}$
$Pr(C_3) = \frac{1}{8} + \frac{1}{8} = \frac{1}{4}$
$Pr(C_4) = \frac{1}{12} + \frac{1}{24} + \frac{1}{12} + \frac{1}{24} = \frac{1}{4}$
$P(K) = \frac{1}{n}$

Figure 1: Our whiteboard solution

(b) How many runs does this procedure run in expectation before you have received the business cards of all employees?

Answer:
Finding the expected number of attempts required, to collect the business card of every employee, is a problem identical to the *coupon collectors problem.*
This is a geometric sum, where the random variable is dependent on how many distinct business cards have already been collected. The probability of getting a new business card is $p = \frac{n-i+1}{n}$ where n is the number of employees and i is the number of distinct business cards collected. The expectation of a geometric sum, is $\frac{1}{p}$ where p is the probability. So using linearity of expectations, we can find the expectation of the random variable X, representing the problem, by summing the expectations of $\{x_1, ..., x_i\}$, which represents the expected time spent in each step. The expectation of a geometric sum is $\frac{1}{p}$, in this case, as mentioned earlier, the probability p for this problem is $p = \frac{n-i+1}{n}$:

$$\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} \frac{n}{n-i+1} = n \cdot \sum_{i=1}^{n} \frac{1}{n-i+1}$$

When moving the n, out of the summation, we are left with a sum, that can be re-interpreted by reversing the order of the sum as:

$$\mathbb{E}[X] = n \cdot \sum_{i=1}^{n} \frac{1}{n-i+1} = n \cdot \sum_{i=1}^{n} \frac{1}{i}$$

$\sum_{i=1}^{n} \frac{1}{i}$ is the n'th harmonic number H_n. The harmonic numbers can be bounded by $log(n)$ (we are not going to show how this is the case). If we put this value in instead of the sum, we get the following:

$$\mathbb{E}[X] = n \cdot log(n)$$

Which is the expected number of attempts needed to collect the business cards from all the employees.

---

4. Let $G = (V, E)$ be a graph with n vertices. For $k \in \mathbb{N}$, a *walk* of length k in G is a sequence of vertices $W = (v_0, \ldots, v_k)$ such that $\{v_i, v_{i+1}\} \in E$ for all $0 \le i < k$. Note that the vertices are not required to be distinct. We say that a walk $W = (v_0, \ldots v_k)$ is a path if the vertices $v_0, \ldots, v_k$ are all distinct.

   (a) Let A be the adjacency matrix of G. For $k \in N$, show that the entry (u, v) of A_k counts the walks of length k from u to v in G. Using this, give an algorithm for counting walks of length in k in time $O(n^\omega)$ when $k = O(1)$.

Answer:
For this assignment, we introduce the example of the undirected graph G and it's adjacency matrix A, representing each edge from u to v.
The result of performing matrix multiplication on A for A^k4 returns the matrix C representing count number of possible walks from u to v.
Example for $k = 2$ and:

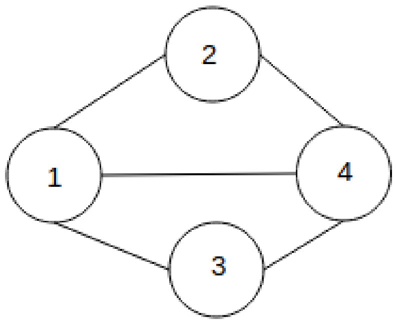

Figure 2: Graph G of size 4

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$A^k = A^2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 3 \end{bmatrix}$$

As shown above, we can use the adjacency matrix A, to count the number of walks of length k from u to v, by taking A^k. This matrix multiplication can be done in $O(n^\omega)$ time, by using Strassen's algorithm for matrix multiplication, or one of its faster successors. In order to then count the total number of walks of length k in the graph G, we need to sum up the entries in the A^k matrix. This gives us the number of walks in k, including both directions of a walk if these are different (meaning that the start and end point of the walk has to be distinct from each other). This summation takes $O(n^2)$ arithmetic operations. $O(n^2)$ is however dominated by $O(n^\omega)$, so the final running time for counting the number of walks of length k in G is $O(n^\omega)$.

   (b) Let $c : V(G) \to \{1, \ldots, k\}$ be a function. We view c as a function that assigns a color to each vertex of G. A walk $W = (v_0, \ldots, v_k)$ is *colorful* if $c(v_i) \ne c(v_j)$ for all $0 \le i < j \le k$. That is, every color appears exactly once in W. Give an algorithm for counting colorful walks of length in k in time $O(n^\omega)$ when $k = O(1)$. (Hint: Since we assume that $k = O(1)$, you can enumerate all possible orderings of the k colors along a path in $k! = O(1)$ time. For any ordering of colors, count paths that respect this ordering. To do this, rather than taking A^k, define appropriate matrices $B_{i,j}$ for $1 \leq i, j \leq k^2$ and take their products.)

Answer:
To count the number of colorful walks in a graph of length k, we first enumerate all possible orderings of the k colors given. For each color sequence, we then construct an adjacency matrix for each consecutive pair of colors, describing only adjacencies between vertices of those 2 colors. So for a color sequence R, G, B, Y, we construct the colored adjacency matrices $B_{R,G}$, $B_{G,B}$ and $B_{B,Y}$. This results in $k-1$ matrices, which are then multiplied together. The entries of the resulting matrix is then summed up, in order to count the number of colorful walks in the graph G, for that specific color sequence. We then repeat this process for each color sequence and sum up all of the results, to obtain the total number of colourful walks in the graph G.

Since finding all possible orderings of k takes $O(1)$ time, and the construction of the $k-1$ color-pair matrices takes $O(n^2)$ time, The running time remains dominated by the matrix multiplication which uses $O(n^\omega)$ time. So this algorithm runs in $O(n^\omega)$ time.

An example of this process for a small graph and $k = 3$ can be seen below:

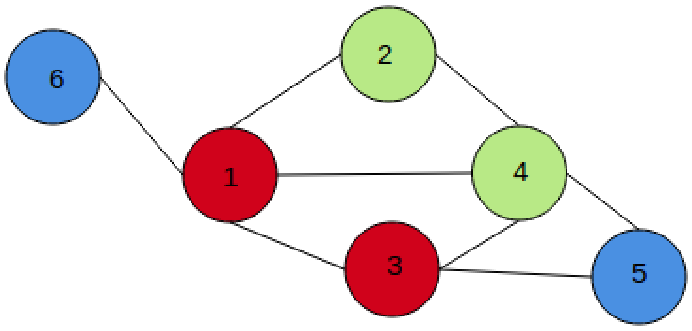

Figure 3: Graph G of size 6

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{R,G} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B_{G,B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{R,G} \cdot B_{G,B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) Let W be any fixed walk of length k in G. We draw a coloring

$$c : V(G) \to 1, \ldots, k$$

uniformly at random.

i. If P is a path, then what is the probability that W is colorful? (Hint: The walk $W = (v_0, \ldots, v_k)$ has k vertices. Count the good colorings on these vertices and divide by the number of possible colorings.)

ii. Using the fact that

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

for any $x \in \mathbb{R}$, show that the above probability is at least $\frac{1}{e^k}$.

iii. If W is not a path, then what is the probability that W is colorful? (Hint: This is a trick question.)

Answer:
Solution (resubmit):
i.

$$Pr(\text{fixed path P is colorful}) = \frac{(k+1)k \cdot (k-1) \ldots}{(k+1)^{(k+1)}} = \frac{(k+1)!}{(k+1)^{(k+1)}} \tag{1}$$

ii.
Continuing from above

$$e^x = \sum_{n=0}^{\infty} \frac{(k+1)^n}{n!} = \frac{(k+1)^{(k+1)}}{(k+1)!} + const$$
$$\geq \frac{(k+1)^{(k+1)}}{(k+1)!}$$

Which leads to

$$\Rightarrow \frac{(k+1)^{(k+1)}}{(k+1)!} \geq \frac{1}{e^{k+1}}$$

iii.
If W is not a path, then there is no probability of it being colorful.

(d) Using the above parts, give a randomized algorithm for testing whether G contains a path of length k. Your algorithm should run in time $O(n^\omega)$ for $k = O(1)$. It should always give the right answer when G has no path, and it should give the right answer with probability at least 0.99 when G does contain a path.

Answer:
Solution (resubmit):
Using the knowledge from above, we can create an randomized algorithm that outputs if a path of length k exist with $P \geq 0.99$.

```
for _ in range($e^k$):
    g = Coloring graph randomly,
    found = check g for colorful path
    if not found:
        return false
return true
```

With repetition we get a succces rate of $\frac{1}{e^{k+1}}$ for the probability that a path exists

---