Test 1 — Purdom

2-27-02 — B403 Analysis of Algorithms

For questions 1–7 consider the following algorithm, which sets i to the location of the first occurrence of the item x in the array $A[1..n]$ (i is set to $n+1$ when x does not occur in the array).

Step 1. Set $i := 1$.
Step 2. If $i > n$ then go to Step 6.
Step 3. If $x = A[i]$ then go to Step 6.
Step 4. i:=i+1;
Step 5. Go to Step 2.
Step 6. Exit.

Assume that steps 1, 4, and 5 need one time unit, steps 2 and 3 need 2 time units, and step 6 needs no time. (These times are for a single execution of the step.)

1. If the first occurence of x is location k, how often is Step 2 done?
2. How many time units are used under the assumptions of question 1?
3. How many time units are used if x is not in the array?
4. Assume that $A[i]$ contains item x with probability p. What is the probability that Step 2 is done at least k times?
5. Under the assumptions of question 4, what is the average time used by the algorithm?
6. What is the limit of the time as n goes to infinity?
7. An alternate algorithm needs time $6k-1$ when x is in the array and time $6k+1$ when x is not in the array. Write down an inequality that is true when the algorithm on the exam is faster than the alternate algorithm. If one algorithm is always at least as fast as the other one for all p and n which are appropriate for this problem, say which one is the fast algorithm. Otherwise say "Depends on n and p".

8. You flip n true coins. If an odd number of them land heads up, you get to keep the ones that land heads up. What is the expected number of coins that you get to keep?

For questions 9–11 consider putting items 1 through k into baskets at random. For each basket and item, item i has probability p of going into the basket, independent of the other items and other baskets.

9. What is the probability that a basket has all k items?
10. What is the probability that a basket has all k items, except for item i (which it can not have)?
11. Suppose we fill b baskets at random. What is the probability that j_0 baskets obey the condition of problem 10, j_1 obey the condition of problem 11 with $i = 1$, j_2 obey the condition of problem 11 with $i = 2$, $\ldots$, and j_k obey the condition of problem 11 with $i = k$? Note that we are asking this set of b baskets to obey a $k + 1$ conditions all at one time.

1. k.
2. $6k-1$.
3. $6n+1$.
4. $(1-p)^{k-1}$ for $1 \leq k \leq n$, $(1-p)^n$ for $k=n$, 0 for $p>n$.
5.

$$\sum_{1\leq k\leq n}(6k-1)p^{k-1}+(6n+1)p^n = 6\frac{(n-1)(1-p)^n - n(1-p)^n + 1 - p}{p^2} + 5\frac{1-(1-p}{p}$$

6.

$$6\frac{1-p}{p^2} + 5\frac{1}{p} = \frac{6-p}{p^2}$$

when $p<1$, 5 when $p=1$.

7. This is the slow algorithm.
8.

$$2^{-n}\sum_{i \text{ odd}} i\binom{n}{i} = n2^{-n}\sum_{i \text{ odd}}\binom{n-1}{i-1} = n2^{-n}\sum_{i \text{ even}}\binom{n-1}{i} = n/4$$

for $n=0$ and $n \geq 2$, $1/2$ for $n=1$.

See below for the last three questions.

For questions 1–7 consider the following algorithm, which sets i to the location of the first occurrence of the item x in the array $A[1..n]$ (i is set to $n+1$ when x does not occur in the array).

Step 1. Set $A[n+1] = x$.
Step 2. Set $i := 1$.
Step 3. If $x = A[i]$ then go to Step 6.
Step 4. i:=i+1;
Step 5. Go to Step 2.
Step 6. Exit.

Assume that steps 2, 4, and 5 need one time unit, steps 1 and 3 need 2 time unit and step 6 needs no time. (These times are for a single execution of the step.)

1. If the first occurence of x is location k, how often is Step 2 done?
2. How many time units are used under the assumptions of question 1?
3. How many time units are used if x is not in the array?
4. Assume that $A[i]$ contains item x with probability p. What is the probability that Step 2 is done at least k times?
5. Under the assumptions of question 4, what is the average time used by the algorithm?
6. What is the limit of the time as n goes to infinity?
7. An alternate algorithm needs time $6k-1$ when x is in the array and time $6k+1$ when x is not in the array. Write down an inequality that is true when the algorithm on the exam is faster than the alternate algorithm. If one algorithm is always at least as fast as the other one for all p and n which are appropriate for this problem, say which one is the fast algorithm. Otherwise say "Depends on n and p".

8. You flip n true coins. If an even number of them land heads up, you get to keep the ones that land heads up. What is the expected number of coins that you get to keep?

For questions 9–11 consider putting items 1 through k into baskets at random. For each basket and item, item i has probability p of going into the basket, independent of the other items and other baskets.

9. What is the probability that a basket has all k items?
10. What is the probability that a basket has all k items, except for item i (which it can not have)?
11. Suppose we fill b baskets at random. What is the probability that j_0 baskets obey the condition of problem 10, j_1 obey the condition of problem 11 with $i = 1$, j_2 obey the condition of problem 11 with $i = 2$, $\ldots$, and j_k obey the condition of problem 11 with $i = k$? Note that we are asking this set of b baskets to obey a $k + 1$ conditions all at one time.

1. k.
2. $4k+1$.
3. $4n+1$.
4. $(1-p)^{k-1}$ for $1 \le k \le n$, $(1-p)^n$ for $k=n$, 0 for $k>n$.
5.

$$\sum_{1 \le k \le n} (4k+1)(1-p)^{k-1} + (4n+1)(1-p)^n = 4\frac{(n-1)(1-p)^n - n(1-p)^n + 1 - p}{p^2} + \frac{1}{}$$

6.

$$4\frac{1-p}{p^2} + \frac{1}{p} = \frac{4-3p}{p^2}$$

when $p > 0$, 5 when $p = 1$.

7. This is the fast algorithm.
8.

$$2^{-n} \sum_{i \text{ even}} i \binom{n}{i} = n2^{-n} \sum_{i \text{ even}} \binom{n-1}{i-1} = n2^{-n} \sum_{i \text{ odd}} \binom{n-1}{i} = n/4$$

for $n \ge 2$, 0 for $n < 2$.

9. $P(k) = p^k$.
10. $Q(k) = (1-p)p^{k-1}$.
11.

$$\binom{b}{j_0, j_1, \ldots, j_k, b - j_0 - \cdots - j_k} P(k)^{j_0} Q(k)^{j_1 + \cdots + j_k} [1 - P(k) - kQ(k)]^{b - j_0 - \cdots - j_k}.$$