Institut for Matematik og Datalogi
Syddansk Universitet

Exam Assignment 2
Algorithms and Probability — 2018

This is the second of two sets of problems (assignments) which together with the oral exam in January constitute the exam in DM551. You must do this assignment individually, not in groups.

The assignment is due at 12:15 on Wednesday, December 12. You may write this either in Danish or English. Write your full name clearly on the first page of your assignment (on the top, if it's not a cover page). Turn it in as a PDF file via Blackboard through your DM551 course. The assignment hand-in is in the menu for the course and is called "SDU Assignment". Keep the receipt it gives you proving that you turned your assignment in on time. Blackboard will not allow you to turn in an assignment late.

Cheating on this assignment is viewed as cheating on an exam. Do not talk with anyone either than Kristine Knudsen or Joan Boyar about the assignment, and do not show your solutions to anyone else. If you have questions about the assignment, come to Joan Boyar or Kristine Knudsen.

Assignment 2

Do the following problems. Write clear, complete answers, but not longer than necessary. Explain all answers, especially those where the answer is very short.

1. Consider the problem of determining if the majority of the elements in an array have the same value (strictly more than $\frac{n}{2}$ elements if the array has n elements. For example $[1, 0, 1, 0, 1]$ has majority element 1, while $[1, 0, 2, 0]$ does not have a majority element.

    (a) Use randomized selection (from Section 13.5 of Kleinberg and Tardos), followed by at most n comparisons to design an expected linear time algorithm for solving this problem.

(b) **procedure Majority-Element**(T, ϵ):

{ Input: Array T of n elements; error probability ϵ }

{ Output: *true* if the majority of elements in T have the same value; *false* otherwise }

```
k = log_2(1/ε)
for i = 1 to k
    j ∈_R [1..n]
    x = T[j]
    count = 0
    for ℓ = 1 to n
        if T[ℓ] == x then count = count + 1
    if count > n/2 then return true
return false
```

Note that the notation $j \in_R [1..n]$ means that j is chosen randomly to be one of the integers between 1 and n inclusive, using the uniform distribution. The notation $==$ compares the two operands for equality.

i. Consider one iteration of the outer **for** loop. Suppose that T contains an element which occurs more than $n/2$ times. Give a lower bound on the probability that **Majority-Element** returns *true* in this iteration.

ii. Using Big-O notation, what is the running time of the algorithm (assuming the comparison for equality takes constant time, as does choosing the random value j)?

iii. Give an upper bound on the error probability of the algorithm. Explain your answer, including explaining when an error can occur.

2. Solve the following linear recurrence relations:

(a) $a_n = 9a_{n-2}$, where $a_0 = 1$ and $a_1 = 3$.

(b) $a_n = 9a_{n-2} - 8n^2$, where $a_0 = 1$ and $a_1 = 2$.

For both problems, check that your results give the same answer as the recurrence does for a_2.

3. These problems concern the Rabin-Karp algorithm for string matching.

(a) Create a family of inputs to the Rabin-Karp algorithm, both a pattern and a text, where the pattern contains at least two different symbols (characters from the alphabet), and the running time of the Rabin-Karp algorithm is $\Omega(mn)$, where m is the length of the pattern and n is the length of the text. The family of inputs should be such that your inputs are parameterized by two variables, one for the pattern and one for the text, and the length of the pattern and text should depend on these variables.

(b) Describe another family of inputs to the Rabin-Karp algorithm, both a pattern and a text, where the expected asymptotic running time of the Rabin-Karp algorithm is $O(n+m)$ and thus asymptotically independent of the length of the pattern, as long as it is not longer than the text. The family of inputs should be parameterized as in the previous problem.

4. What is the prefix function computed by the KMP algorithm for the string $P = abbabbabcabbabc$.

5. Consider a modification of the Gambler's Ruin problem discussed in class, where the gambler makes exactly n bets (where n is not necessarily the amount the gambler started with). Thus, there is no upper limit making him/her stop, and he/she is able to borrow money if the result becomes negative. Thus, it is easier to assume starting with 0 dollars and just consider how much is won or lost. Let p be the probability of winning $+1$ with each bet, and $q = 1 - p$ be the probability of losing (adding -1). Let Z_i be the random variable which has value 1 if there was a win on the ith bet and -1 if there was a loss, and let $S_n = \sum_{i=1}^{n} Z_i$.

   (a) Use the following steps to show that $E[S_n^2] = n$ when $p = 1/2$.

      i. Show that $E[S_n^2] = \sum_{i=1}^{n} \sum_{j=1}^{n} E[Z_i \cdot Z_j]$. Which properties of expectations do you use?

      ii. What is $E[Z_i \cdot Z_j]$ for fixed $i \neq j$ (a number, not a function of p, for $p = 1/2$)?

      iii. What is $E[Z_i \cdot Z_i]$ for fixed i? (a number, not a function of p, for $p = 1/2$)?

      iv. Why is $E[S_n^2] = n$ when $p = 1/2$?

(b) Consider the same questions as in the previous problem for the more general case where $p \neq 1/2$. (Instead of numerical values, you will get functions of p and q.) Show that

$$E[S_n^2] = (n^2 - n)(4p^2 - 4p + 1) + n.$$

(c) In the case where $p = 1/2$, use Chebyshev's inequality to find an upper bound on the probability that $|S_n| \geq 2\sqrt{n}$.