Randomized Algorithms Multiple Choice Test

Sample test: only 8 questions – 24 minutes

(Real test has 30 questions – 90 minutes)

Årskort

Name

Each of the following 8 questions has 4 possible answers of which exactly one is correct. For each question, you may select one or more answers by checking the corresponding boxes. Your test is graded as follows:

- If you select only the correct answer, you receive 2 points.
- If you select 2 answers, one of which is correct, you receive 1 point.
- If you select 3 answers, one of which is correct, you receive 0.4 point.
- if you select no answer or all answers, you receive 0 point.
- If you select only one answer, and it is wrong, you receive -1 point.
- If you select 2 answers, that are both wrong, you receive -1.6 points.
- If you select 3 answers, that are all wrong, you receive -2 points.

Note that perfect answers yield a score of 16 points and random guessing is likely to give you a negative score.

Question 1
What does the randomised Marker algorithm for on-line paging do? You may assume that the size of the cache is k.

- *a* ☐ It works in rounds. Within each round it counts the number of accesses to each page in the cache, and evicts a random page among those with the least number of accesses. With probability $1/k$ it starts a new round and resets all counts to 0.
- *b* ☐ It works in rounds. Within each round it counts the number of accesses to each page in the cache, and evicts a random page among those with the least number of accesses. It starts a new round and resets all counts to 0 when there has been k accesses to the cache.
- *c* ☐ It counts the number of accesses to each page in the cache, and evicts a random page among those with the least number of accesses.
- *d* ☐ It works in rounds. Within each round it marks all accessed pages, and evicts a random non-marked page. It starts a new round and clear all marks when there is no unmarked page available to evict.

Question 2
Define the function M that maps bit-strings into 2×2 matrices as follows

$$M(0) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad M(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

and

$$\text{for } x = x_1 \ldots x_n \in \{0,1\}^*, \quad M(x) = \prod_{i=1}^{n} M(x_i).$$

It is known that $M(x) = M(y)$ if and only if $x = y$. It is also known that all four entries in the matrix $M(x)$ are numbers in the range $0 \ldots 2^n$.
We make a randomized equality test on bit strings as follows. We select a random prime p from a set of primes $\mathcal{P}$ and on input x, y, we compute $M_p'(x) = M(x) \bmod p$ and $M_p'(y) = M(y) \bmod p$. Then we compare the two values, $M_p'(x)$ and $M_p'(y)$.
We want at most a probability 1/2 for making an error, i.e. for finding $M_p'(x) = M_p'(y)$ when in fact $M_p(x) \neq M_p(y)$.
What is the least size of $\mathcal{P}$ among the following choices that guarantee this bound on the error probability?

- *a* ☐ $|\mathcal{P}| = n$
- *b* ☐ $|\mathcal{P}| = 2^n$
- *c* ☐ $|\mathcal{P}| = 2 \log n$
- *d* ☐ It is not possible to get that bound on the error probability for any value of $|\mathcal{P}|$

Question 3

What does Adleman's theorem state?

a ☐ If a Boolean function has a randomized, polynomial-sized circuit family, then it also has a non-randomized, polynomial-sized circuit family

b ☐ There are problems that can be solved by a randomized polynomial time algorithm for which there is no deterministic polynomial time algorithm

c ☐ Every problem that can be solved by a randomized polynomial time algorithm can also be solved by a deterministic polynomial time algorithm

d ☐ There are Boolean functions that can be solved by a randomized polynomial sized circuit family that cannot be solved by a non-randomized polynomial-sized circuit family.

Example

Algorithm:	Randomised Quicksort (RQ)
Input:	$L = [a_1, \ldots, a_n]$, a list of distinct numbers
Output:	L sorted in ascending order
Method:	(sketch only)

If $|L| \leq 1$ then return L otherwise:

1. Select e randomly from L using the uniform distribution
2. Split $L' = L - \{e\}$ into the two sublists
   $L_1 = [a_i \in L | a_i < e]$
   and
   $L_2 = [a_i \in L | a_i > e]$
   by comparing e to each element of L'.
3. Recursively, sort L_1 and L_2.
4. Return
   sorted-$L_1 \cdot [e] \cdot$ sorted-L_2

Let $b_1, \ldots, b_n$ be defined as the output of the sorting algorithm, i.e. $b_1 < \cdots < b_n$ and $b_1, \ldots, b_n$ is a permutation of $a_1, \ldots, a_n$. Define random variables X_{ij} with values 0 and 1 such that $X_{ij} = 1$ if and only if we compare b_i and b_j during the execution of algorithm RQ (when splitting in line 2 in some recursive incarnation of the algorithm)

Question 4

What is the expected number of comparisons made by algorithm RQ

a ☐ $\mathtt{E}[2\sum_{i<j}(1 - X_{ij})]$

b ☐ $\mathtt{E}[\sum_{i \neq j} X_{ij}]$

c ☐ $\mathtt{E}[2\sum_{i \neq j}(1 - X_{ij})]$

d ☐ $\mathtt{E}[\sum_{i<j} X_{ij}]$

Question 5

In the analysis of RQ, what is $\Pr(X_{ij} = 1)$? Assume that $i \neq j$

a ☐ $\Pr(X_{ij} = 1) = 2/(j + i)$

b ☐ $\Pr(X_{ij} = 1) = 1/(j + i)$

c ☐ $\Pr(X_{ij} = 1) = 2/(|j - i| + 1)$

d ☐ $\Pr(X_{ij} = 1) = 1/(|j - i| + 1)$

Question 6

Let $h_1, \ldots, h_4$ be hash functions from $M = \{0, 1, 2, 3, 4\}$ to $M = \{0, 1\}$ defined as follows

x	$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$
0	1	0	1	0
1	0	1	0	1
2	1	0	0	1
3	0	1	1	0
4	1	1	0	0

Which of the following sets H is 2-universal?

a ☐ $H = \{h_1, h_3\}$
b ☐ $H = \{h_4\}$
c ☐ $H = \{h_2\}$
d ☐ $H = \{h_1, h_2, h_3, h_4\}$

Question 7

Consider the Valiant-Brebner randomized oblivious two-phase scheme for permutation routing, where the bit fixing algorithm is used to route packages in each of the two phases. What is the expected length of the total path travelled by a single package under this scheme applied on the d dimensional hypercube?

a ☐ d
b ☐ 2^d
c ☐ $2 \log d$
d ☐ $2d$

Question 8

If we have to generate a single uniformly random number from $\{1, 2, 3\}$, how many random bits do we need for that?

a ☐ We need 2 random bits
b ☐ No upper bound can be given
c ☐ We need $\log_2 3$ random bits
d ☐ We need 3 random bits