OHJ-2016 Utilization of Data Structures

Exam 13.5.2008

It is forbidden to use any written material such as cheatsheets, books, lecture notes etc. Electrical devices (calculators, cell phones, computers, etc.) may not be used during the exam.

- Answer in English unless explicitly permitted otherwise.
- Answer briefly and clearly - the answers are not graded based on their length.
- Incorrect answers do not reduce points. However, irrational or inconsistent answer may get 0 points, even if some part of it is correct.

0. Leave five empty rows at the bottom of the first page of the first paper. ($\frac{1}{4}$ p)

1. Explain the following terms briefly (at most 3 lines per item) (Finnish answers are OK):

   a) dynamic programming (1 p)

   b) input enhancement (1 p)

   c) algorithm (1 p)

2. `Algorithm` has started at node s and is currently processing node u. Node v is adjacent to node u and is grey. Explain (i) what—if anything—this situation means, (ii) what is known about the distance of v from s and (iii) what is the definition of "distance" if `algorithm` is

   a) depth-first search for undirected graphs (1 p) / for directed graphs (1 p)

   b) breadth-first search for undirected / directed graphs (2 p)

   c) Dijkstra's algorithm for undirected / directed graphs (2 p)

   If you need to make further assumptions or to use other markings besides s, u and v, explain them clearly.

3. a) Draw a "worst-case" red-black tree that is as much out of balance as possible with the keys 3, 3, 7, 7, 11, 15, 19, 19, 22, 55. Draw *every* node from root to the leaves or explain unambiguously (and briefly) the tree's structure. Would some other set of 10 keys make an even more out-of-balance red-black tree possible? (2 p)

   b) Array [9, 4, 7, 3, 3, 6, 1, 2, 8] should represent a heap, but it got broken somehow. Why is it not valid? What is the most likely explanation to how it got broken? How is it fixed? (2 p)

   c) An application is performing poorly. The source seems to be a function that is implemented using insertion sort. Give one possible explanation to the poor performance. (1 p)

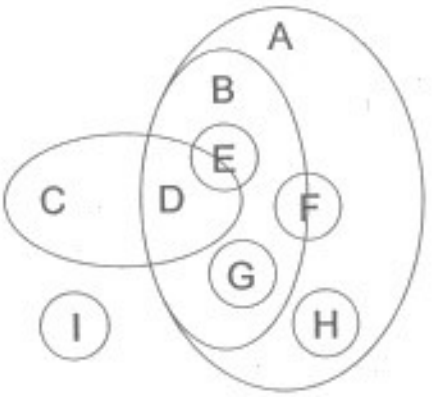
4. The diagram below represents sets and their relationships. For example, B is a (strict) subset of A, sets C and B overlap and sets F and G are distinct.

   Place the items in the following lists in the diagram by associating each item with a set (write e.g. "A" or "C") or an element of a set (write e.g. "in A" or "in C"). Use more complicated expressions (e.g. "A ∩ C" or "A - B") or textual comments **only** if the items can not be placed otherwise.

Example:	Exam question	Your answer
	a) $\Phi(n^2)$, $\Psi(\lg n)$, $\Lambda(mn)$, f	a) X, in Z, Y, Å
	b) $\Phi(n)$, $\Psi(n \lg n)$, $\Lambda(\ln n)$, g	b) Z, M, in X, in R

a) Four items (2 p)
   $O(n)$,
   $\Omega(n)$,
   $\Theta(n)$,
   exec. time of inserting into a skip-list

b) Six items (3 p)
   $\Omega(3n)$,
   $\Theta(n \lg n)$,
   $O(n^*)$,
   $n^3 + 3n - 6$,
   $n^6 - n^4 + n^2 - 1$,
   exec. time of inserting into a skip-list



c) Six items (3 p)
   $\Theta(n - 19)$,
   $\Theta(n^3)$,
   $O(7)$,
   $\Omega(10 \log_{10} n)$,
   $\Omega(\frac{n+1}{5})$,
   $\Omega(2 \log_2 n)$

5. Tell us your opinion of the course. What should be changed or improved and what should be kept as it is? Did you find TRAKLA useful? How often did you attend the lectures? Write your answer on a separate page, alone without your name, but turn it in together with your exam answers. ($\frac{3}{4}$ p)

   NOTE: There is one more real question on the next page!

6. a) What does `exam_func` do? What is its time and space complexity with O and Ω notations? Which algorithm design principle/strategy is used? The time complexity is a bit difficult to determine, so the boundaries do not have to be very tight. For space complexity do not consider the space used for the input itself. Vector `T` is empty for the top-level call. (2 p)

   b) The problem could be solved much much more efficiently. What makes this case such a bad target for using this particular algorithm design strategy? There are two main reasons; one has to do with the similarity of (some) equally good solution candidates and the other with decomposition to subproblems. How could you utilize the first property to significantly improve the algorithm? You you must preserve the recursive structure and must not increase memory consumption. (2 p)

   c) What strategy or strategies could you use if memory consumption was not a top priority? What if the algorithm did not have to be recursive? What is the time complexity for the latter case? (2 p)

   d) **Answer either to both of b) and c) or only to d).**
   The programmer forgot to document one important restriction on the input. What is it and what does `exam_func` do if it is violated? (0.5 p)
   How would you improve the algorithm's execution time? Estimate how significant the improvement is. (0.5 p)

   e) Examining the usage of `exam_func` you discover regularity in `B`. Elements of `B` fulfill a condition $b_i = k_i b_{i-1}$ for $i > 1$ where b_i is the ith largest integer in `B` and k_i is a positive integer. Note that the value for b_1 can still vary and that there can be duplicate values.
   The condition allows using a simple approach that is highly efficient. What is it? Why does it work for this problem? What is the basic idea of the algorithm? What is its time and space complexity? (2 p)

You do not need to invent anything new. Answers showing some understanding of the problem are rewarded. Answers without reasoning (i.e. guesses) get no points. You may answer in Finnish.

```
uint exam_func(uint A, vector<uint> &B, vector<uint> &T) {
  uint x = A;
  for(int i = 0; i < B.size(); i++) {
    if (B[i] <= A) {
      vector <uint> D;
      uint tmp = exam_func(A -B[i], B, D);
      if (tmp < x || (tmp == x && D.size() < T.size() -1)) {
        x = tmp;
        T.assign(D.begin(), D.end());
        T.push_back(i);
      }
    }
  }
  return x;
}
```