

## CS 260 Evening Class, Take-Home Test 3

### CS 260 Take-Home Test Honesty Statement

I am fully aware that I am not allowed to consult anybody about the problems assigned for the take-home exam except the instructor and the TAs, and that during the work on test problems I may only access the textbook and the materials available through CS 260 course website. By making a submission of my answers I acknowledge that I followed all of the above.

Name:

Signature:

Solve all three problems. Each problem counts for 3 points.

1. Consider the following algorithm for sorting an array segment  $A[0..n-1]$ . In the first step the algorithm performs the bubble-up operation on the range  $[0..n-1]$  and it places the smallest item on position 0. In the second step it performs the bubble-down operation on the range  $[1..n-1]$  and it places the largest item on position  $n-1$ . In the third step it performs bubble-up operation on the range  $[1..n-2]$  and it places the second smallest item on position 1. In the fourth step it performs the bubble-down operation on the range  $[2..n-2]$  and it places the second largest item on position  $n-2$ . The algorithm continues alternating bubble-up and bubble-down operations until the range consists of a single field.
  - (i) What is the number of swap operations performed in the worse case?
  - (ii) What is the average number of swap operations performed by this algorithm?

Provide justifications to your answers.

2. A sorted list of  $n$  strings is given. Adapt the binary search algorithm in order to determine whether a given string  $x$  is a member of this list. Assume that the list allows constant time access to all of its fields by an integer index. What is the time complexity of your algorithm in terms of  $n$  and the length of  $x$ ? Justify your answer.
3. What is the maximal number of swap operations performed during the process of selecting consecutive minima during the execution of the heap-sort algorithm? In your count skip the swap operations performed during the process of creating the initial heap. Assume that the input array has size  $n$ .
  - (i) Express your answer in the form of a sum.
  - (ii) Remove the symbol of summation and analyze the order of growth.

Hint: Make use of the formula:

$$1*2^1 + 2*2^2 + 3*2^3 + \dots + N*2^N = (N-1)*2^{N+1} + 2.$$

```
procedure pushdown(first, last: integer);
  var
    r: integer;
  begin
    r := first;

    while r <= last div 2 do
      if last = 2*r then begin
        if A[r].key > A[2*r].key then
          swap(A[r],A[2*r]);
          r := last
        end
      else
        if A[r].key > A[2*r].key and
          A[2*r].key <= A[2*r+1].key then begin
          swap(A[r],A[2*r]);
          r := 2*r
        end
        else if A[r].key > A[2*r+1].key and
          A[2*r+1].key < A[2*r].key then begin
          swap(A[r],A[2*r+1]);
          r := 2*r+1
        end
      else
        r := last
      end
    end; { pushdown }

procedure heapsort;
  var
    i: integer;
  begin
    for i := n div 2 downto 1 do
      pushdown(i,n);
    for i := n downto 2 do begin
      swap(A[1],A[i]);
      pushdown(1,i-1)
    end
  end; { heapsort }
```