

CS 260 Review 3

1. A sorting algorithm based on swap operations of neighboring positions of an input array is given. Is it possible to determine that the number of swap operations occurring in its worst case is bounded below by a polynomial function applied to the size of the array? If the answer is yes, provide an optimal polynomial, i.e. a polynomial that provides a bound from below for all such algorithms and for which equality occurs for some of them? In both cases of your answer “yes” and “no” provide a justification.
2. What is the maximal number of swap operations performed during the process of creating the initial heap during the execution of the heap-sort algorithm? Assume that the input array has size n .
 - (i) Express your answer in the form of a sum.
 - (ii) Remove the symbol of summation from your formula.
 - (iii) 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; { 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 }

```

3. Suppose we have a set of words, i.e. strings of letters a - z , whose total length is n . Show how to sort these words in $O(n)$ time.
4. Find and solve recurrence relations describing:
 - (i) the number of moves needed for moving n disks in the Towers of Hanoi problem,
 - (ii) the number of bit operations needed to multiply integers if we assume that two multiplications are enough to accomplish the process of reduction of size from n to $n/2$.
5. The entries a_{ij} of matrix A are computed according to the formula

$$\begin{aligned}
 a_{ij} &= 1 \text{ for } i=1, j>1, \\
 a_{ij} &= 0 \text{ for } i>1, j=1, \\
 a_{ij} &= (a_{i-1,j} + a_{i,j-1})/2 \text{ for } i>1, j>1.
 \end{aligned}$$

- (i) What number of operations $+$ is necessary to compute n anti-diagonals of the matrix A ? Provide an exact answer. Do not include a_{11} .
- (ii) Express numbers a_{ij} in terms of the entries of Pascal's triangle.