CS 204 Midterm Wedneday November 7 2001

Show all work. no calculators Time: 50 minutes. Marks: 20

1. (2 marks) Show all changes after `rankUnion(3,9)` with compressed find.

```
        1  2  3  4  5  6  7  8  9 10
rank   [3  0  2  0  1  0  0  2  1  0]

rank   [                            ]  <- changed ranks

parent [1  1  1  3  3  5  5  8  8  9]

parent [                            ]  <- changed parents
```

2. (2 + 1 marks) Heapsort is performed on A below. (i) Show A as indicated:

```
A [3 2 9 4 6 8 1 7 5]

A [                ] <- after BuildHeap

A [                ] <- after next MaxHeapify
```

(ii) The maximum number of key comparisons performed by a MaxHeapify of 1023 keys is:

3. (3 marks) Quicksort is performed on the initial A above. Draw the quicksort recursion tree. Label each node with the elements of the subarray being sorted.

4. (3 marks) For n keys, give the asymptotic running time:

	average case	worst case	best case
insertion sort			
heapsort			
quicksort			
selection sort			

5. (2 marks) $M[x, y]$ is the minimum number of scalar multiplications needed to perform the chain matrix multiplication $A_x \times \ldots \times A_y$. Determine $M[2, 5]$.

```
    0  1  2  3  4  5  6         2   3   4   5   6
d [10 10  3  4  5  2  7]  M 1[300 420           ]
                            2[    120 210       ]
                            3[         60  64   ]
                            4[             40 96]
                            5[                70]
```

6. (2 marks) $A(n)$ is the average number of key comparisons performed on a quicksort of n keys, assuming each of the $n!$ possible input permutations is equally likely. Explain in detail why the following holds.

$$A(n) = n - 1 + \frac{1}{n}\left(\sum_{j=1}^{n} A(j-1) + A(n-j)\right)$$

7. (2 + 1 + 2 marks)

```
proc zip(n)
  if n < 4 then return n
  else
    sum <- 0
    for j <- 1 to n-1 do
      sum <- sum + zip(j)
    return sum
```

(i) Give a recurrence relation for $P(n)$, the number of nodes in the recursion tree of `zip(n)`.

(ii) Give the simplest, most accurate, asymptotic expression for $P(n)$. Justify.

(iii) Using pseudocode, write the most efficient procedure `zap(n)` you can which always returns the same value as `zip(n)`.