

Maths 3467 quizzes

3: Thursday 30/10/13 due Thu 14/11

Please be sure you understand what is wanted. If in doubt, send e-mail. Collaboration is permissible.

(1) Of the four sorting methods discussed, which fit the ‘comparison model’? Your answer should mention their estimated runtimes.

Answer. Only the ‘bucket sort’ method does not fit the model. The other three have $\Omega(n \log n)$ worst-case behaviour as predicted by the model, indeed quicksort has $\Omega(n^2)$ worst-case behaviour.

(2) Consult the web notes for a detailed presentation of heapsort and also mergesort. Is heapsort stable? If so, explain why, and if not, show with an example that it is unstable.

Answer. No. Apply it to sorting 1, 2a, 2b which has two occurrences of 2.

1 2a 2b is already heapified. Removing 1,
we are obliged to replace it by 2b, which is the next min chosen.
2b 2a
so the items are output in the order
1 2b 2a.

Do the same for mergesort.

Answer. No.

With input

1a 1b

Because merge swaps items of equal value, mergesort outputs

1b 1a

Note. Mergesort as presented in the notes is deliberately unstable. A small change to the merge procedure will make it stable.

(3) Simulate the construction of the overlap function for the following pattern:

a b a b b a b b a b a b b a

0 1 2 3 4 5 6 7 8 9 10 11 12 13
a b a b b a b b a b a b b a

m j k pattern pattern ovlp[k]
 [k+1] [j+1]

	0			k=0: -1
-1		b	a	
	1			k=1: -1
		a	a	
0				k=2: 0
	2			
		b	b	
1				k=3: 1
	3			
		b	a	
-1				k=4: -1
	4			
		b	a	
		a	a	
0				k=5: 0
	5			
		b	b	
1				k=6: 1
	6			
		b	a	
-1				k=7: -1
	7			
			a	
		a	a	
0				k=8: 0
	8			
		b	b	
1				k=9: 1
	9			
		a	a	
2				k=10: 2
	10			
		b	b	

```

3      11
      11      k=11: 3
      b      b
4      12
      12      k=12: 4
      a      a
5      13
      13      k=13: 5

```

Final result

```

0  1  2  3  4  5  6  7  8  9  10 11 12 13
a  b  a  b  b  a  b  b  a  b  a  b  b  a
-1 -1  0  1 -1  0  1 -1  0  1  2  3  4  5

```

(4) Simulate the KMP algorithm with pattern

a b a b b

and text

a b a b a b a b b a

```

0  1  2  3  4
a  b  a  b  b      pattern
-1 -1  0  1 -1      ovlp

0  1  2  3  4  5  6  7  8  9      text
a  b  a  b  a  b  a  b  b  a

m  n  j  k  pat j+1  text k+1
5  10
      -1
    -1
      a      a
    0
      0
      b      b
    1
      1
      a      a
    2
      2

```

```

          b      b
3
  3
    b      a
1
  a
2
  4
    b      b
3
  5
    b      a
1
  a
2
  6
    b      b
3
  7
    b      b
4 break
return 4

```

(5) In the Aho-Corasick generalisation of the KMP algorithm, backlinks are installed in the TRIE (after all strings have been inserted) as follows.

- It should have been mentioned that different children of a node are labelled with different letters.
- Nodes of the trie are processed in breadth-first order. This is accomplished using a first-in-first-out queue.
- The backlink for the root is undefined and never consulted.
- Initially, the children of the root are queued with their backlinks set to the root.

```

while ( queue nonempty )
{
  take node u from front of queue
  for ( all letters a labelling children of u )
  {
    let v be the a-child of u
    let w be the back-link of u

    while ( w not root and w has no a-child )
      w = backlink ( w );
  }
}

```

```

    if ( w has an a-child x )
        backlink ( v ) = x
    else
        backlink ( v ) = root

    append v to the queue
}
}

```

Simulate it on the trie below. Take nodes in the order of their labels, which is breadth-first. Keep track of the nodes u, v, w, x , noting when their back-links are installed.

u	v	w	x	
1				
	4			
		0	2	backlink(4)
2				
	5			
		0	3	backlink(5)
3				
	6			
		0		backlink(6)=0
4				
	7			
		2	5	backlink(7)=5
	8			
		2		
		0		backlink(8)=0
5				
	9			
		3	6	backlink(9)=6
6				
	10			
		0		backlink(10)=0
	11			
		0		backlink(11)=0
7				
	12			
		5		
		0		backlink(12)=0
8	no children			
9				
	13			
		6	10	backlink(13)=10

10
 14 0 backlink(14)=0
 11 15
 0 backlink(15)=0
 12 no children
 13
 16
 10 14 backlink(16)=14
 14
 15
 16 no children

