

Några svar till TDDI16 Datastrukturer och algoritmer

2014-10-28

Följande är lösningsskisser och svar till uppgifterna på tentan. Lösningarna som ges här ska bara ses som vägledning och är oftast inte tillräckliga som svar på tentan.

1. 1 (B)
- 2 (G)
- 3 (L)
- 4 (D)
- 5 (F)

2. (a)

		5		
	4		8	
3		6	10	
			25	

- (b) 5, 4, 3, 8, 6, 10, 25

3. (a) 19 26 32 35

För att sätta in en nod i en binär heap sätter vi in den som nytt sidta löv och låter den bubbla upp. Alltså är nycklarna 19, 26, 32, 35 och 38 de enda som kunnat flytta på sig. Men, den sista nyckeln som satts in kan inte ha varit 38 eftersom 35 då skulle ha behövt vara den gamla rotnoden (och detta skulle bryta mot heapordningen så vänter barn till rotnoden är 37).

- (b) 14 17 19 34 35 36 37

Jämförelserna är 35-37, 19-37, 34-36, 19-36, 14-17, 19-17.

4. |12|9|13| | | |45| | |33|10| |

5.

0	1	2	3	4	5	6	7	8	9
	31	22		14	435	346	157	568	
		212			5			8	
		102							

- | | | | | | | |
|-----|-----|----|-----|-----|-----|-----|
| 102 | 212 | 22 | 31 | | 157 | 568 |
| 5 | 14 | | 435 | 346 | | |
| 8 | | | | | | |

- | | | | | | |
|----|-----|-----|-----|-----|-----|
| 5 | 102 | 212 | 346 | 435 | 568 |
| 8 | 157 | | | | |
| 14 | | | | | |
| 22 | | | | | |
| 31 | | | | | |

6.
 1. Sortera varje rad med heap-sort.
 2. För varje tal i rad 0, från störst till minst, använd binärsökning för att kontrollera om det uppträder på de andra $N - 1$ raderna.
 3. Returnera det första talet som uppträder på alla N rader.

Exekveringstiden växer som $N^2 \log(N)$ med flaskhalsar i setg 1 och 2. Korrekthet följer eftersom det största talet måste finnas på rad 0. Undersökningen av rad 0 från störst till minst försäkrar att vi hittar det *största* gemensamma talet.

7. Huvudidén är att använda en ordnad symboltabell för att lagra strängarna på stacken, där den i :te strängen som satts in är värdet associerat med heltalsnyckeln i . Om symboltabellen implementeras med t.ex. ett AVL-träd kan vi lätt hitta och ta bort strängen som senast sattes in i stacken (dess nyckel är den största i trädet). Vi kan också enkelt slumpa fram ett heltal och snabbt ta bort motsvarande sträng.

```
class LeakyStack {
public:
    void push(string item) {
        st.insert(counter++, item);
    }

    string pop() {
        string item = st.max();
        st.deleteMax();
        return item;
    }

    void leak() {
        int r = Random.uniform(st.size());
        st.delete(r);
    }

private:
    int counter = 0;
    AVLTree<int, string> st;
};
```