

# Eksamen H2019 ADS101

## Algoritmer og dastrukturer for spill

9/12/19

**Tillatte hjelpemidler:** Filer og programmer på egen pc, lærebok og notater, kursets Canvas-rom med lenker. **Alle deloppgaver teller like mye.**

### 1

(teori) Gitt tallene 17, 14, 5, 7, 12, 1, 16, 29, 13.

- Tegn opp et binært søketre etter at tallene er satt inn i rekkefølge.
- Tegn opp treet etter at tallet 14 er slettet.
- Tegn opp et AVL-tre med de samme tallene satt inn i rekkefølge.

### 2

(teori) Gitt en array med tallene 17, 14, 5, 7, 12, 1, 16, 29, 13, 4, 8, 18, 22, 2.

- Anta at tallene skal sorteres med en umodifisert rekursiv quicksort, hvor pivotelementet velges ved  $(\text{left} + \text{right})/2$ . Her er left og right indeks til henholdsvis første og siste element i array. Sett opp en tabell som viser alle bytter i første kall på den rekursive quicksort-funksjonen. Forklar hva som er oppnådd i dette funksjonskallet.
- Anta nå at tallene skal sorteres med mergesort. Sett opp en tabell som viser hvordan hele sorteringen foregår, helt til du har en ferdig sortert array. Forklar med tekst i tillegg.
- Hva er kompleksiteten til disse algoritmene? Regn ut for begge tilfellene ovenfor.

### 3

Følgende C++ kode er gitt:

```
using namespace std;

struct Test {
    int key;
    std::string s;
    bool operator == (const Test& t2) const { return key == t2.key; }
};

namespace std {
    template<>
    class hash<Test> {
    public:
        size_t operator() (const Test& t) const {
            return t.key % 7;
        }
        bool operator() (const Test& t1, const Test& t2) {
            return t1.key == t2.key;
        }
    };
}

int main(int argc, char *argv[])
{
    hash<Test> hashtabell;
    unordered_set<Test> uordnet_sett;
```

- a) (programmering) Skriv kode for å sette inn poster med nøkler 12, 13, 20, 21, 1, 2, 3, 4, 5, 6 i uordnet\_sett i opplistet rekkefølge.
- b) (teori) Tegn en figur som viser hvordan postene blir plassert i uordnet\_sett.
- c) (teori) Hvor mange buckets er det i uordnet\_sett etter at postene er satt inn? Hva blir fyllingsgraden?

## 4

(programmering) I denne oppgaven skal du bruke kode som er gitt til slutt i oppgaven, uten endringer.

- a) Implementer funksjonene *Node::settinn\_kant()*, *Graf::settinn\_node()* og *Graf::finn\_node()*. I sistnevnte funksjon skal du benytte den interne strukturen til å søke etter en node med gitt navn. Funksjonen skal returnere en peker til noden dersom den fins, og nullptr ellers.
- b)
  1. Implementer funksjonen *Graf::settinn\_kant()*
  2. Lag en testgraf i *main()* med noder {**A**, **B**, **C**, **D** og **E**} og kanter {**AB(1.0)**, **AC(2.0)**, **BC(2.0)**, **CD(3.0)**, **DE(1.0)**, **AE(5.0)**, **CE(4.0)**}
- c) implementer funksjonen *mst()* slik at den returnerer kostnaden for et minste spennetre i grafen.

```
#include <iostream>
#include <list>
#include <queue>
using namespace std;

struct Kant;
struct Node {
    char m_navn;
    bool m_besokt;
    std::list<Kant> m_kanter;
    Node(char navn) : m_navn(navn), m_besokt(false) { }
    void settinn_kant(const Kant &kant);
};

struct Kant {
    float m_vekt;
    Node* m_tilnode;
    Kant(float vekt, Node* tilnode) : m_vekt(vekt), m_tilnode(tilnode) { }
    bool operator > (const Kant& k) const { return m_vekt > k.m_vekt; }
};

struct Graf {
    std::list<Node*> noder;
    Graf() { }
    Node* finn_node(char navn);
    void settinn_node(char navn);
    void settinn_kant(char fra_navn, char til_navn, float vekt);
    float mst();
};
```

*Slutt på oppgaven.*