

Fasit til
Eksamen H2020 ADS101
Algoritmer og datastrukturer for spill

??/??/21

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

1

(teori og programmering, 25%)

Gitt tallene 2.71828, 1.732, 1.414, 3.1416, 9.81.

- a) Tegn et binært søketre av tallene, satt inn i rekkefølge som ovenfor.
- b) Tegn en min-heap av tallene, satt inn i rekkefølge som ovenfor.
- c) Tegn en max-heap av tallene, satt inn i rekkefølge som ovenfor.
- d) Skriv et C++ program som bygger en max-heap med de samme tallene, gjør en pop operasjon og skriver ut roten etter pop.

2

(programmering, 25%)

Følgende C++ kode er gitt:

```
class BinaryNode {
public:
    BinaryNode(double data=0.0);
    void insert(double data);
    void stigende() const;
    void avtakende() const;

private:
    double m_data;
    BinaryNode* m_left;
    BinaryNode* m_right;
};

BinaryNode::BinaryNode(double data)
    : m_data{data}, m_left{nullptr}, m_right{nullptr} { }

void BinaryNode::insert(double data) {
    if (data < m_data) {
        if (m_left)
            m_left->insert(data);
        else
            m_left = new BinaryNode(data);
    } else if (data > m_data) {
        if (m_right)
            m_right->insert(data);
        else
            m_right = new BinaryNode(data);
    }
}
```

- a) Skriv koden til funksjonen stigende() slik at den skriver ut noderes verdier i stigende rekkefølge når den kalles av roten i et tre.
- b) Skriv koden til funksjonen avtakende() slik at den skriver ut noderes verdier i avtakende rekkefølge når den kalles av roten i et tre.
- c) Skriv koden til en funksjon som summerer alle noderes verdier.
- d) Skriv et testprogram hvor du setter inn verdiene 2.71828, 1.732, 1.414, 3.1416, 9.81 i denne rekkefølgen. Test funksjonene fra a, b og c.

3

(programmering, 30%)

I denne oppgaven skal du bruke kildekode fra løsningsforslag til obligatorisk oppgave 3 - punkt 4 (Quadtree). Hele dette prosjektet er tillatt brukt. Hvis du vil, kan du gjøre alle objektvariabler public.

Gitt en scene avgrenset av $-4 \leq x \leq 4$, $-4 \leq y \leq 4$ og fem gameObject instanser med selvvalgte navn, og koordinater $(-2.0, -2.0)$, $(1.0, 3.0)$, $(3.5, 1.5)$, $(3.2, 1.2)$, $(3.4, 1.4)$.

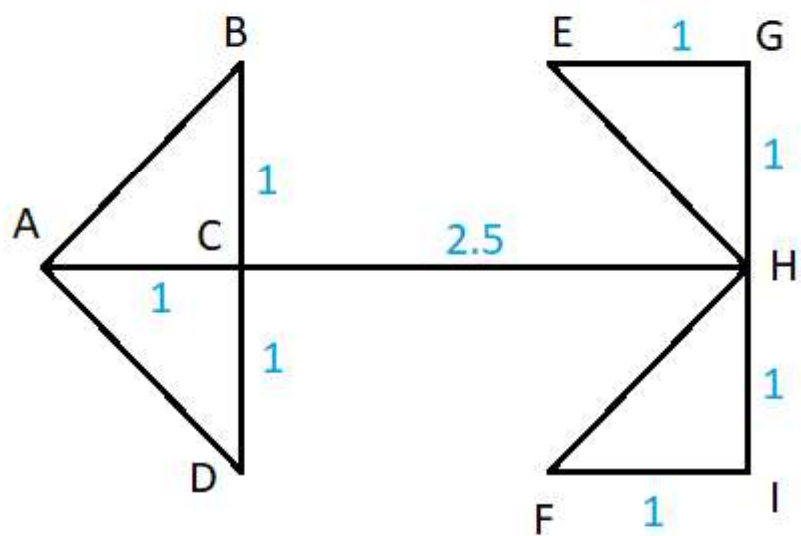
- a) Definer i main() fire Vector2d objekter som angir hjørnene til scenen, og opprett fem GameObject instanser (objekter).
- b) Lag et Quadtree objekt som rommer hele scenen. Del deretter opp subtreet **m_no** en gang til, og subtreet **m_so** til dette enda en gang.
- c) Legg inn (referanser til) de fem objektene av type GameObject i Quadtreeet.
- d) Skriv en funksjon void Quadtree::printSub(const Vector2d& m) const; som skriver ut alle objektene til et quad som inneholder punktet m, når funksjonen kalles av roten.
- e) Skriv en funksjon void printAll() const; som skriver ut alle objektene i quadtreeet når den kalles av roten. Test disse to funksjonene i main().

4

(teori, 20%)

Figur 1 viser en graf med navngitte noder og kantenets vekt. La C være en startnode og E en sluttnode. Bestem korteste vei fra startnode C til sluttnode E, og skriv opp prioritetskøa med vekt/lengde underveis, når du bruker Dijkstra's algoritme.

Slutt på oppgaven.



Figur 1: Graf