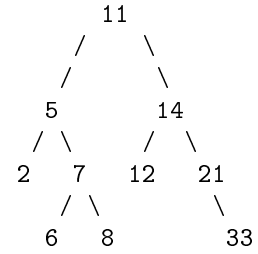
Eksamen H2018 ADS101
Algoritmer og dastrukturer for spill

10/12/18

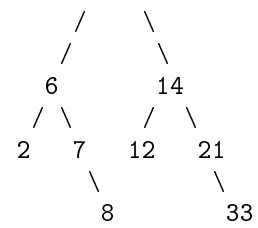
1 Binærtre fasit

(teori - vekt $\frac{1}{8}$)

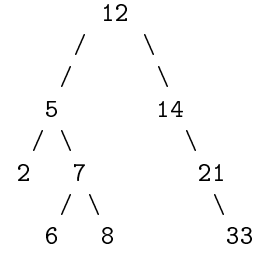
a1) Tallene innsatt i binært søketre:



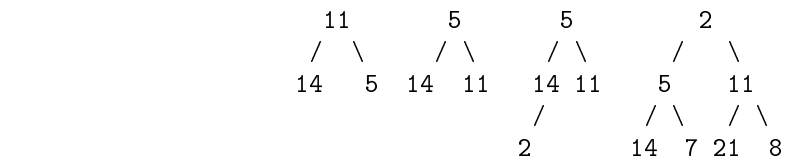
a2) Tallet 5 slettet:

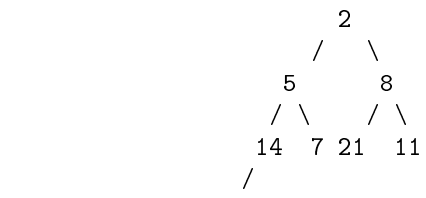


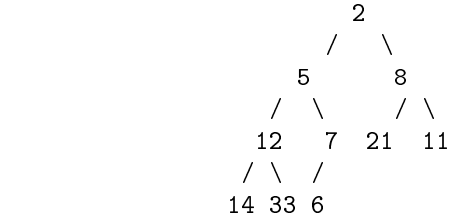
a3) Tallet 11 slettet:

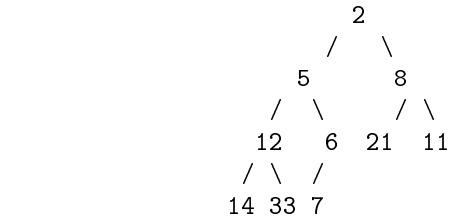


b1)

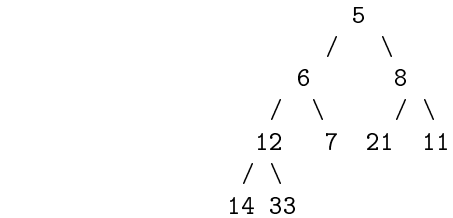

11 5 5 2
14 5 14 11 14 11 5 11
2 14 7 21 8

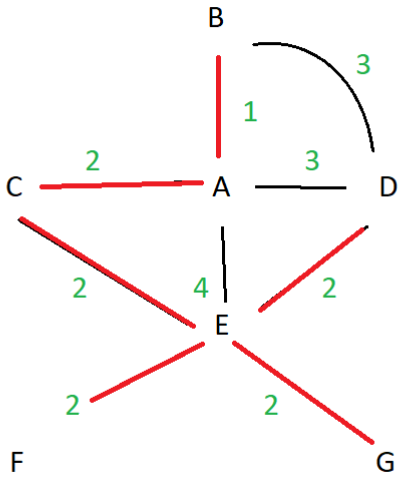

2
5 8
14 7 21 11
12


2
5 8
12 7 21 11
14 33 6


2
5 8
12 6 21 11
14 33 7

Etter pop:


5
6 8
12 7 21 11
14 33



Figur 1: Oppgave 2a) med mst (2b) markert.

2 Graf fasit

(teori - vekt $\frac{1}{8}$)

b) mst (11).

			ED(2)			
			EF(2)	EF(2)		
AB(1)	AC(2)	CE(2)	EG(2)	EG(2)	EG(2)	
AC(2)	AD(3)	AD(3)	AD(3)	AD(3)	AD(3)	AD(3)
AD(3)	BD(3)	BD(3)	BD(3)	BD(3)	BD(3)	BD(3)
AE(4)	AE(4)	AE(4)	AE(4)	AE(4)	AE(4)	AE(4)

c) Korteste veg BAEF(7), BACEF(7), BDEF(7).

d) Warshall's algoritme. Setter opp matrise A hvor nodenes navn svarer til rader og kolonner og 1 betyr kant, 0 ingen. Boolsk sum A*A*A gir sykluser med tre kanter osv. Alternativt kan man skrive om dybde-først/topologisk sortering og hvordan dette kan gjøres.

				BACE(5)
				BADE(6)
	BAC(3)	BAD(4)	BAE(5)	BAEG(7)
	BAD(4)	BAE(5)	BACE(5)	BAEF(7)
BA(1)	BAE(5)	BACE(5)	BADE(6)	BAED(7)

Her poppes to lengre veger til E, deretter BAEG og så BAEF(7).

3 Sortering fasit

- For å klare å måle tiden for små arrayer, bør klokka startes før første sorteringsrunde og stoppes etter siste runde.
- Det kreves mer enn bare å sette opp resultatene i en tabell, se nedenfor.

Flettesortering er $O(n \log n)$. Da finnes en konstant c slik at sorteringstiden

$$f(n) = c \cdot n \log n \Leftrightarrow c = \frac{f(n)}{n \log n}$$

Tilsvarende gjelder for innstikksortering som er $O(n^2)$. I tabellene nedenfor er c regnet ut for ulike n.

n	10	100	1000	10000
$tid(ns)$	1	9	217	1927
$n \cdot \log_{10} n$	10	200	3000	40000
c	0.1	0.045	0.072	0.048

Vi ser her at for de tre største datasettene ligger c i intervallet (0.5, 0.7). Tilsvarende utregning for innstikksortering gir c i intervallet (0.08, 0.09) for de to største datasettene.

n	10	100	1000	10000
$tid(ns)$	0	14	816	91288
n^2	1	10^4	10^6	10^8
c		0.0014	0.00816	0.009

Tallene vil selvsagt variere for hver enkelt besvarelse. Andre formuleringer, figurer etc kan være like bra.

4

```
Binaer_tre* Binaer_tre::test_scene() {
    Binaer_tre* stjerne = new Binaer_tre{"stjerne", Vektor3d{0.5, 0.0, 0.5}, nullptr, null
    Binaer_tre* vindu = new Binaer_tre{"vindu", Vektor3d{2.0, 0.0, 1.0}, stjerne, nullptr}
    Binaer_tre* doer = new Binaer_tre{"doer", Vektor3d{1.0, 0.0, 0.0}, nullptr, vindu};
    Binaer_tre* hus = new Binaer_tre{"hus", Vektor3d{2.0, 2.0, 0.0}, doer, nullptr};
    stjerne = new Binaer_tre{"stjerne", Vektor3d{0.0,0.0,3.0}, nullptr, nullptr};
    Binaer_tre* tre = new Binaer_tre{"tre", Vektor3d{8.0, 1.0, 0.0}, stjerne, hus};
    return new Binaer_tre{"scene", Vektor3d{0.0, 0.0, 0.0}, tre, nullptr};
}
void Binaer_tre::skriv(const Vektor3d &pos) {
    std::cout << navn << ", ";
    Vektor3d v = posisjon+pos;
    std::cout << v.x << ", " << v.y << ", " << v.z << std::endl;
    if (venstre) {
        venstre->skriv(v);
    }
    if (hoyre)
        hoyre->skriv(pos);
}
```

```
scene, 0, 0, 0
        tre, 8, 1, 0
              stjerne, 8, 1, 3
hus, 2, 2, 0
        doer, 3, 2, 0
        vindu, 4, 2, 1
              stjerne, 4.5, 2, 1.5
```