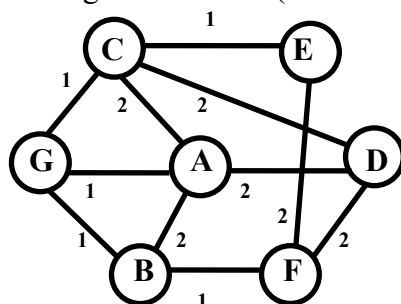


Obligatorisk oppgave nr.5

IDATG2102 – Algoritmiske metoder, høsten 2023

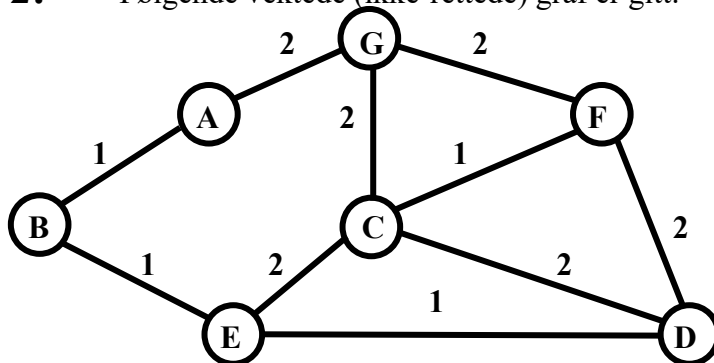
Frist: 31.oktober 2023 kl.11:00 (må overholdes) via Blackboard

Oppgave 1: Følgende vektete (ikke-rettede) graf er gitt:



Vi bruker nabomatrise, og starter i node G. **Tegn opp minimums spenn-treet (MST)** for denne grafen, etter at koden i EKS_31_MST.cpp er utført/kjørt. **Skriv også opp innholdet i/på fringen etterhvert** som konstruksjonen av MST pågår. **NB:** Husk at ved lik vekt så vil noden sist oppdatert (nyinnlagt eller endret) havne først på fringen ift. andre med den samme vekten.

Oppgave 2: Følgende vektete (ikke-rettede) graf er gitt:



Vi bruker nabomatrise. Koden i EKS_32_SP.cpp utføres/kjøres på denne grafen. **Hvilke kanter er involvert i korteste-sti spenn-treet fra noden B til alle de andre nodene?** **Skriv også opp innholdet i/på fringen etterhvert** som koden utføres. **NB:** Husk at ved lik vekt så vil noden sist oppdatert (nyinnlagt eller endret) havne først på fringen ift. andre med den samme vekten.

Oppgave 3:

Vi har en litt spesiell form for retted graf, der alle nodene har *maksimalt en etterfølger*, men *gjerne flere forgjengere*. Dvs. fra hver node går det *maksimalt en kant ut*, men *gjerne flere kanter inn*. Blant slike grafer finnes bl.a: enkelt-linkede lister, enkle sykliske strukturer (løkker) og rotrettede trær (der alle «barn» har *en* peker opp til «mora», mens «mor» har altså *ikke* en eller flere pekere ned til barna sine) + mange flere ulike grafer.

Nodene er representert ved:

```
struct Node {
    int ID; // Nodens ID/key/data.
    Node* etterfølger; // Peker til etterfølgeren, evt. nullptr.
    Node(const int id) { ID = id; etterfølger = nullptr; }
};
```

Det er N noder. Den globale arrayen: `Node* gGraf[N+1]` har direkte pekere til *alle* grafens noder (vi bruker indeksene 1- N) i en eller annen *helt tilfeldig* rekkefølge.

Lag funksjon `bool erRotrettedTre()` som returnerer `true/false` til om grafen *virkelig* er et rotretted tre eller ei. Beskriv også tankegangen for algoritmen/funksjonen.

NB: Det skal *ikke* innføres flere globale data eller `struct`-medlemmer enn angitt ovenfor. Det skal heller *ikke* brukes andre hjelpestrukturer (utover den angitte `gGraf`) - som f.eks. array, stakk, kø eller liste.

Hele obligen (både tegninger, tekst og kode for funksjonen) leveres som en PDF innen fristen via emnets rom i Blackboard.

FrodeH