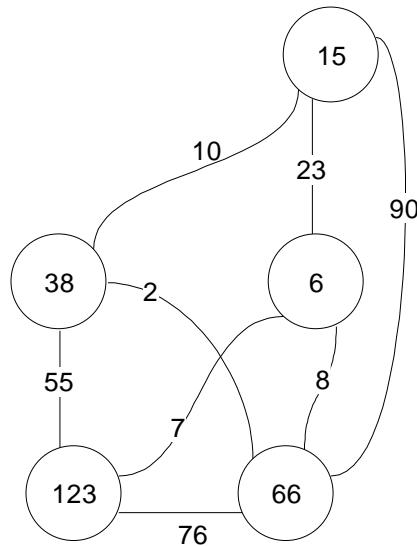


Opgaver onsdag den 19. marts

Opgave 1

- a) Brug den udleverede implementering af en graf (*graph.zip*) til at opbygge en graf svarende til:



- b) Test din graf fra a) ved at udskrive grafen.
For knuden med værdien 6 skal du udskrive: graden (degree), nabo knuderne, de tilstødende kanter.
- c) Skriv en metode i en App klasse, der givet en graf som input, kan finde den største værdi indeholdt i grafens knuder (Det kan antages at elementerne i kunderne er heltal). Afprøv metoden på grafen opbygget i opgave a). Resultatet af at køre metoden på denne graf, skulle gerne være 123.

Opgave 2

I den udlevede *EdgeListGraph* er metoderne til at fjerne knuder og kanter fra grafen ikke implementeret. Metoden til at afgøre om to knuder mangler også. Programmer metoderne:

```
/**
 * Remove the specified vertex from the graph.
 * Pre: The vertex is in the graph, and the vertex has no incident
edges.
 */
public void removeVertex(V vertex);

/**
 * Remove the edge between the specified vertices from the graph.
 * Pre: The vertices are vertices in the graph,
 * and The graph has an edge between the vertices.
 */
public void removeEdge(V v, V u);

/**
 * Return true, if the specified vertices are neighbors.
 * Pre: The vertices are vertices in the graph.
 */
public boolean areAdjacent(V v, V u);
```

Opgave 3

Færdiggør implemenationen af klassen `AdjacencyListGraph`, som implementerer interfacet `Graph` som en *naboliste*:

```
public class AdjacencyListGraph<V> implements Graph<V>
```

Bemærk: Du har fået udleveret en klasse `EdgeListGraph`, som implementerer interfacet `Graph` som en *kantliste*. En hel del af koden fra `EdgeListGraph` kan genbruges i implementationen af `AdjacencyListGraph`. (I Graf-noten findes klassediagrammer for de to implementationer af `Graph`-interfacet.) Hvis en metode kan gøres mere effektiv ved at anvende en knudes naboliste (i stedet for listen over alle kanter), så skal det gøres.

Opgave 4 (? Afhængig af tiden)

Færdiggør implemenationen af klassen `AdjacencyMatrixGraph`, som implementerer interfacet `Graph` som en *nabomatrice*:

```
public class AdjacencyMatrixGraph<V> implements Graph<V>
```