

Testfall för labb 3

DD2350 ADK

Uppdaterad 15 oktober 2021

Dessa testfall är till för att hjälpa er med testning och avlusning av era program. I praktiken (i verkligheten) kommer sådana här testfall normalt inte att finnas tillgängliga, så en viktig förmåga är då att själv kunna konstruera informativa test.

För att klara labben räcker det med att Kattis accepterar er lösning. Ni måste inte köra dessa testfall. Kattis kommer dock inte att ge någon återkoppling om vad som är fel med ert programmet så ni rekommenderas att initialt sätta dessa fall som mål.

In- och utdatafiler kan hittas under katalogen

`/afs/kth.se/misc/info/kurser/DD2350/adk21/lab3/testfall`

Notera dock att det typiskt kan finnas fler än en lösningar till dessa problem och att en direkt jämförelse med exempelutdatafilerna kan vara missvisande. Kattis är mer förlåtande och accepterar samtliga korrekta lösningar.

1 Reducera bipartit matchning till maxflöde

Problem 1.1

Det här problemet låter er testa att ni har gjort en korrekt reduktion från bipartit matchning till maxflöde. Ni skall inte köra flödeslösaren på detta problem utan bara skriva ut er maxflödesinstans och jämföra med den som visas här. Figurerna kan skilja sig åt, speciellt om ni inte använt er av exakt den reduktion som har gått igenom i kursen.

Indata

Se figur för visualisering av in- och utdata.

Filnamn:

tillflodetest.indata.

2 3

4

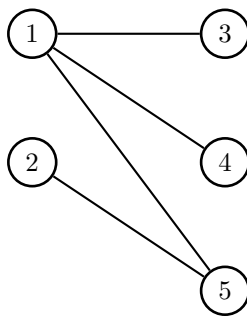
1 3

1 4

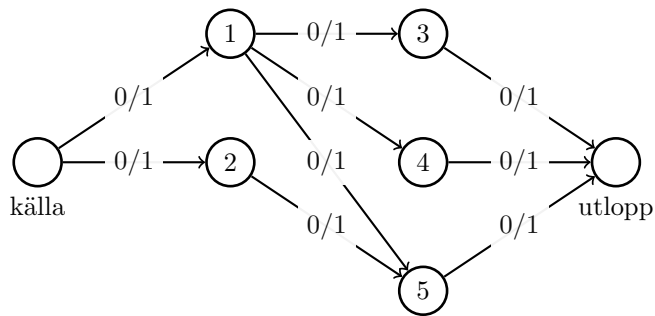
1 5

2 5

Figur 1: Indata och exempelutdata för Problem 1.1.



(a) Indata



(b) Exempelutdata.

Problem 1.2

Detta är en fortsättning till Problem 1.1. Nu förväntas uppgiften lösas i helhet och ge en maximal matchning som utdata.

Indata

Filnamn:

kapacitetstest.indata

2 3

4

1 3

1 4

1 5

2 5

Exempelutdata

Filnamn:

kapacitetstest.utdata

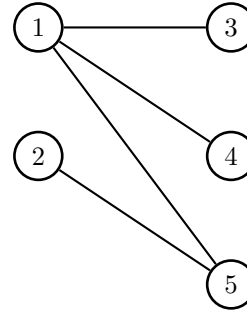
2 3

2

1 3

2 5

Figur 2: Indata för Problem 1.2.



Problem 1.3

Följande problem testar att ert program kan hantera tomma noder och formaterar utdata korrekt. Ert program bör producera utdata som är *identiskt* med det här angivna.

Indata

Filnamn:

formattest.indata

4 2

1

3 5

Utdata

Filnamn:

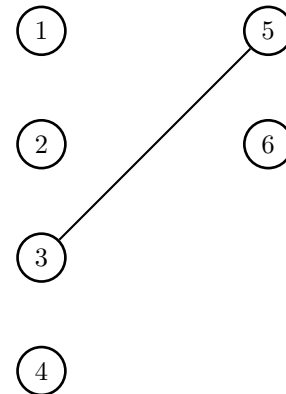
formattest.utdata

4 2

1

3 5

Figur 3: Indata för Problem 1.3.



2 Lös flödesproblemet

Problem 2.1

Det här problemet testar den typ av flödesproblem som ni kommer skapa i steg 3.

Indata

Filnamn:
bipartittest.indata

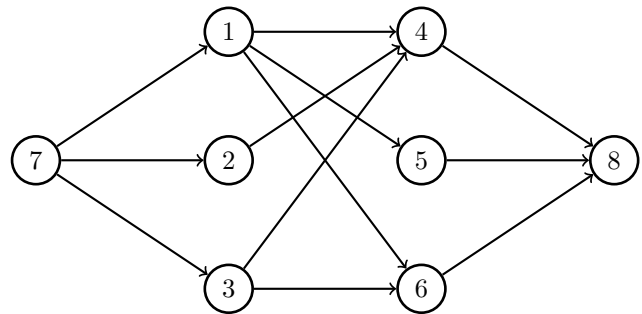
```
8
7 8
12
1 4 1
1 5 1
1 6 1
2 4 1
3 4 1
3 6 1
7 1 1
7 2 1
7 3 1
4 8 1
5 8 1
6 8 1
```

Exempelutdata

Filnamn:
bipartittest.utdata

```
8
7 8 3
9
1 5 1
2 4 1
3 6 1
4 8 1
5 8 1
6 8 1
7 1 1
7 2 1
7 3 1
```

Figur 4: Indata för Problem 2.1. Samtliga vikter är 1.



Problem 2.2

För detta problem bör ni skriva ut antalet iterationer, d.v.s. antalet utvidgande stigar (*augmenting paths*) som programmet behöver för att terminera. En korrekt lösning med breddenförstökning kommer att hitta en lösning med endast en utvidgande stig.

Indata

Filnamn:

`bfstest.indata`

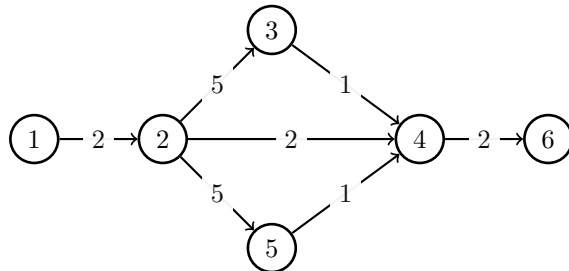
```
6
1 6
7
1 2 2
2 3 5
2 4 2
2 5 5
3 4 1
4 6 2
5 4 1
```

Exempelutdata

Filnamn:

`bfstest.utdata`

```
6
1 6 2
1 2
2 4
4 6
```



Figur 5: Indata för Problem 2.2.

Problem 2.3

Följande exempel testar att ni lägger till och utnyttjar kanter i restflödesgrafen.

Indata

Filnamn:

`residualtest.indata`

```
8
1 5
10
1 2 10
2 3 11
3 4 8
4 5 9
2 7 3
6 4 13
1 6 12
6 7 6
7 8 10
8 5 10
```

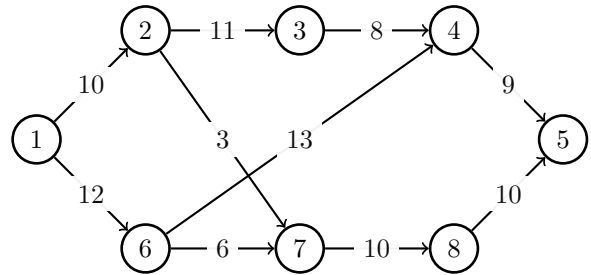
Exempelutdata

Filnamn:

`residualtest.utdata`

```
8
1 5 18
10
1 2 6
1 6 12
2 3 3
2 7 3
3 4 3
4 5 9
6 4 6
6 7 6
7 8 9
8 5 9
```

Figur 6: *Indata för Problem 2.3.*



3 Kombinera steg 1 & 2

Problem 3.1

Det här är matchningsvarianten av ett tidigare problem.

Indata

Filnamn:

matchingstest.indata

3 3

6

1 4

1 5

1 6

2 4

3 4

3 6

Exempelutdata

Filnamn:

matchingstest.utdata

3

3

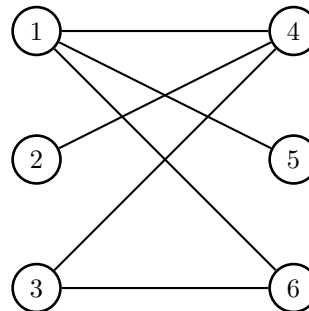
3

1 5

2 4

3 6

Figur 7: Indata för Problem 3.1.



Problem 3.2

Ett sista, lite maffigare, exempel att testa när ni har fått allt annat att fungera.

Indata

Filnamn:

maffigttest.indata

```
5 6
11
1 6
1 7
2 6
2 7
3 7
4 7
4 8
4 9
4 11
5 9
5 11
```

Exempelutdata

Filnamn:

maffigttest.utdata

```
5 6
4
2 6
3 7
4 9
5 11
```

Figur 8: Indata för Problem 3.2.

