

IN3130 Exercise set 6

Exercise 1

Solve Exercise 14.4 in the textbook (B&P) (and sketch a data structure for Exercise 14.5).

Exercise 2

Solve Exercise 14.6 in the textbook.

Exercise 3

Assume $|X| = |Y|$. Then show that, if we have found a subset S of X with $|\Gamma(S)| < |S|$, we can also easily find a subset of T of Y with $|\Gamma(T)| < |T|$.

Exercise 4

Question 4.a

Show that, for general graphs, any “node cover” will never have fewer nodes than there are edges in any matching. A “node cover” is a subset NC of nodes that “covers” all the edges. That is: all edges have at least one of its end nodes in NC .

Question 4.b

Look at some examples with bipartite graphs, and observe that in such graphs you can always find a matching and a node cover of the same size. (It is in fact not difficult to prove this by looking at the situation when the Hungarian algorithm stops after having built alternating trees from all unmatched nodes in X , and no augmenting path has been found. The above fact can be used to prove that a certain match is as large as possible, also for cases with $|X| \neq |Y|$).

Question 4.c

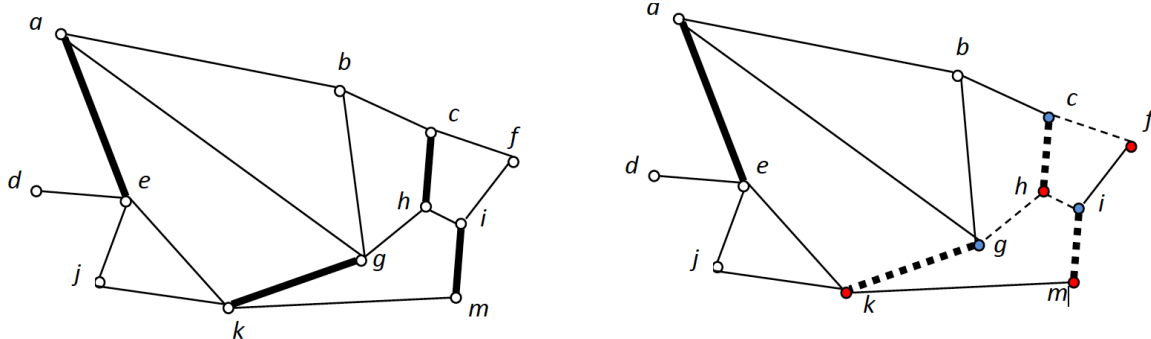
Find an example showing that, in general graphs, one cannot always find a node cover and a matching of the same size.

Exercise 5

We are given the following graph G , and a matching M . You shall use the maximum matching algorithm for general graphs to find a maximum matching for G , by starting with M . Start with node f as the root, then look at the edge $f-c$ getting also $c-h$ into the tree. Then look at edges $h-g$ and $h-i$, which will both increase the tree by two nodes each.

Which nodes are now red and blue (assuming that the root f is red)?

Then look at the unmatched edge out of m . What will happen then? Proceed with choices so that you end up finding an augmenting path between d and f (even though one between b and f or j and f is closer by). Show the resulting matching after you have "used" this augmenting path. Finally, decide whether this matching can be increased further.



Exercise 6

To study the max flow algorithm, go through the example in Figure 14.9 in detail (B&P). See introduction at the bottom of page 439. Note that there are many typos in these graphs in early editions of the book, but most of them should now be corrected. Also note that the first graph in the left column is N (and not N_f), and that in the right column step 6 has the final flow, while the last graph is N itself with the (original) capacities, and where the cut is displayed with dotted edges.

Known typos in early editions of the textbook are:

- Step 1: Edge 4-7 in N_f should be dotted.
- Step 2—7: Edge 4-7 should be reversed in all N_f s.
- Step 2: Inner edges in the flow graph should be removed.
- Step 2: Edge 0-3 in N_f should not be dotted.
- Step 7: Vertex 5 in N should have a double circle, and an edge 2-5 with flow 1 should be added to the flow graph.
- Step 7: The sets should be $X = \{0,1,2,3,5\}$, and $Y = \{4,6,7\}$.

The figure with typos corrected is included at the end of this document.

Exercise 7 (Question 7.c – 7.f can be left to the students)

Study figure 14.10 on page 444 of the text book (B&P). (Note that there are typos in at least some editions of the book: The edge (x_1, y_2) in the upper graph should be removed.) We now look at the duality between finding a maximum matching in the upper graph, and finding a maximum flow in the lower network (graph).

Question 7.a

Look at the following lemma, and explain why it is correct (Hint: This has also been commented on in the lectures, and it relies on the way the algorithm works):

Lemma In a network with integer capacities one can always find a flow that is both maximum and integer, and the Ford-Fulkerson-algorithm will always find such a flow.

In other words: If the capacities are integer, we never have to split a flow so that for instance $\frac{1}{2}$ goes down one edge and $\frac{1}{2}$ down another to achieve a maximum flow. This means that if all capacities are 1, we get a maximum flow for the network with either full (1) or no (0) flow in each edge. Such a flow induces a subset of the edges: those with full flow.

Question 7.b

Use the lemma to explain that finding a maximum matching in the upper graph in Figure 14.10 is the same as finding a maximum flow in the lower network.

Question 7.c

Assume that you in Figure 14.10 have the matching $\{(x_2, y_1), (x_4, y_3), (x_5, y_5)\}$, and show what flow f this corresponds to in the lower network.

Question 7.d

Draw $N(f)$ (the f -derived network) for the flow from 7.c and check that looking for an f -augmenting path from s to t in this graph corresponds to looking for a (matching) augmenting path in the upper graph, with the given matching.

Question 7.e

Use an f -augmenting path found (for instance (x_1, y_1, x_2, y_4) in the graph and $(s, x_1, y_1, x_2, y_4, t)$ in the network) to augment the matching/flow, and check that these operations are duals of each other. Verify that you end up in the situation shown in the lower network in figure 14.10 (where flows are indicated).

Question 7.f

Draw $N(f)$ for this new flow, and show that the flow is a maximum flow by showing a cut with this capacity (4). Then use the method from Exercise 4 above to find a vertex cover of four vertices covering all edges in the upper graph, thereby showing that the matching is a maximum matching. Finally show how the cut and this vertex cover are related.

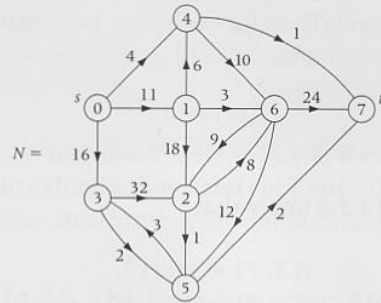
Exercise 8 (If you have time)

Show that the following three conditions on undirected graphs are equivalent:

- The graph is bipartite
- The graph is two-colourable
- The graph has no odd cycles

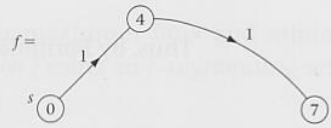
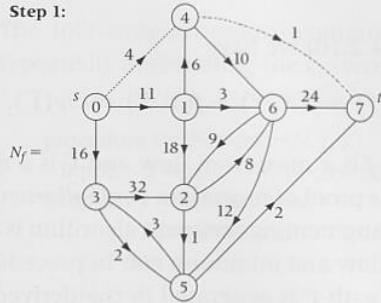
FIGURE 14.9
Action of the Edmonds-Karp algorithm for a sample capacitated network N .

Original flow network N with capacities c , and initial flow $f \equiv 0$:

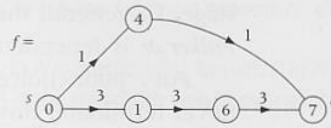
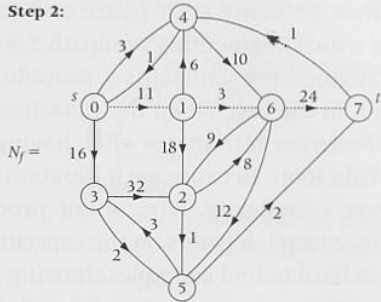


$f \equiv 0$

Step 1:



Step 2:



Step 3:

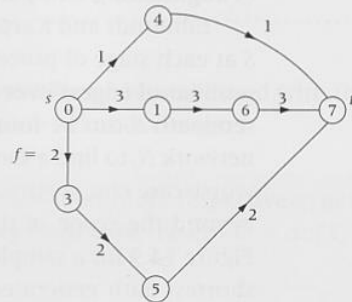
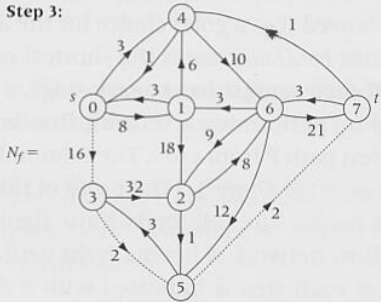
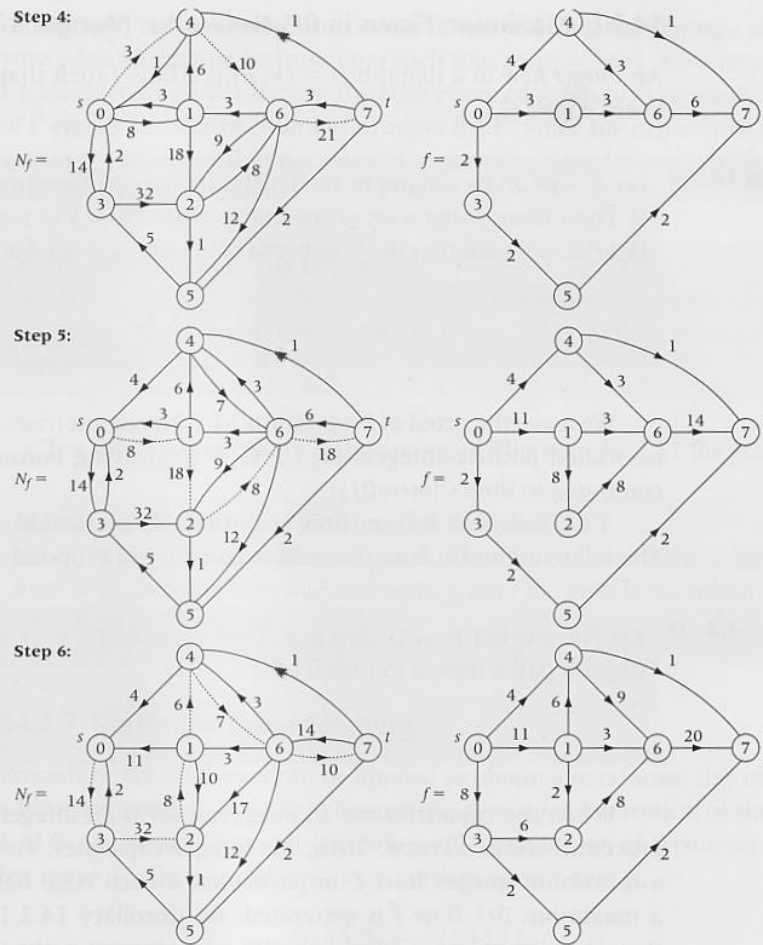
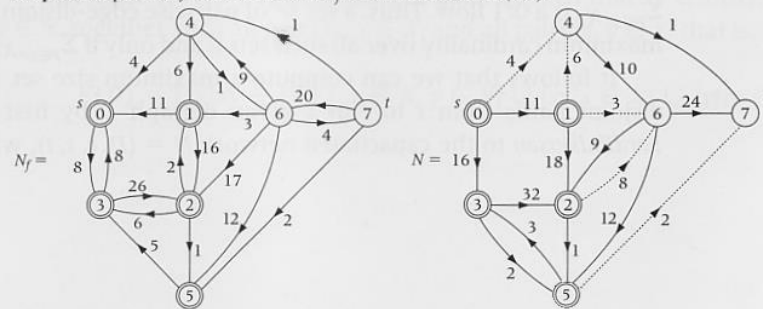


FIGURE 14.9
Continued



There are no more augmenting semipaths. The final flow f has value 23.

Step 7: Compute the f -derived network N_f and minimum cut $cut(X, Y)$.



The set $X = \{0, 1, 2, 3, 5\}$ of vertices that are accessible in N_f from the source s (marked with \odot) and the set $Y = \{4, 6, 7\}$ of vertices that are not accessible from s determine a cut $\Gamma = cut(X, Y)$ of capacity $c(X, Y) = 4 + 6 + 3 + 8 + 2 = 23$. Hence, we have $val(f) = 23 = cap(\Gamma)$, so that f is a maximum

[end]