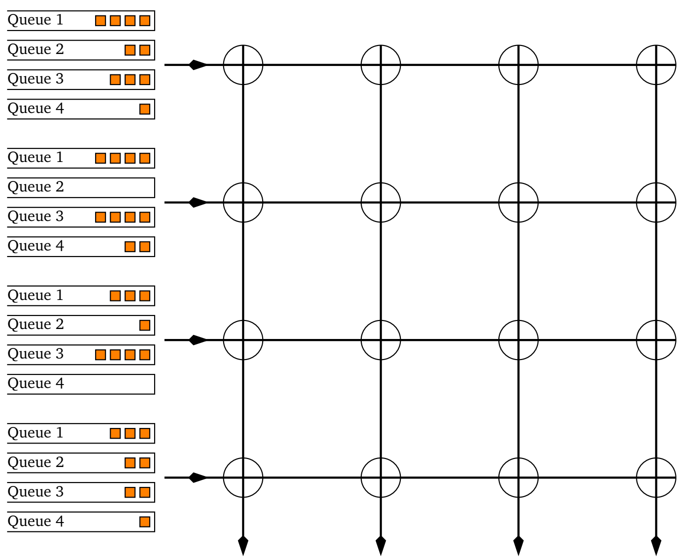
Assignment 5

CSCI 4113/6101
Instructor: Norbert Zeh
Due: Nov 18, 2025, 11:59pm

One way to implement a network router is as a crossbar switch. There are n input wires and n output wires. At their intersections, there are switches that can be turned on and off to create a connection between the wires. Each packet of data arriving on one of the input wires is labelled with the output wire it should be sent to. These packets are collected into queues based on the output wire they should be sent to:



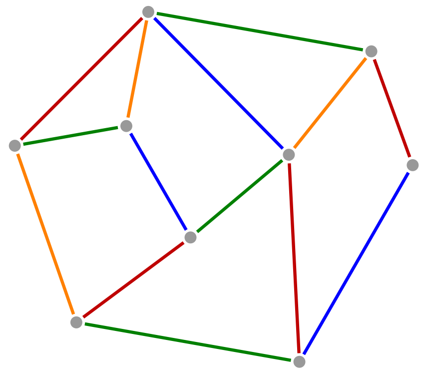
In each time step, the router can send multiple packets from input wires to output wires, but it cannot send more than one packet arriving on the same input wire in the same time step, and it cannot send more than one packet to the same output wire in the same time step (because no wire can be connected to more than one other wire at any time). To maximize the throughput of the router, we would like to send many packets at the same time in each time step.

Now, in the real world, this is an online problem: while the router makes a decision on which packets to send in the current time step, more packets arrive on the input wires. Here, we consider the offline version of this problem: The queues of the input wires are prefilled with packets to be sent to the output wires, each labelled with the output wire it should be sent to; no new packets arrive while the algorithm runs. Our goal is to send all packets from the input wires to the correct output wires in as few time steps as possible subject to the constraint that no input wire can send more than one packet per time step and no output wire can receive more than one packet per time step.

Formally, we represent each packet as a pair (i_j, o_j), where i_j is the input wire on which the packet arrives, and o_j is the output wire to which the packet is to be sent. We are given a set of packets $P = \{(i_1, o_1), \ldots, (i_m, o_m)\}$. Our goal is to partition P into subsets $P_1, \ldots, P_t$ such, for all $1 \le h \le t$, no two packets in P_h arrive on the same input wire and no two packets in P_h are to be sent to the same output wire. We call such a partition $\mathcal{P} = \{P_1, \ldots, P_t\}$ a *valid routing* of the packets in P. It is an *optimal routing* of the packets in P if $t = |\mathcal{P}|$ is minimal among all valid routings of the packets in P.

Question 1

A valid edge colouring of a graph G is an assignment of colours to the edges of G so that the edges incident to each vertex all have distinct colours:



An *optimal* edge colouring is a valid edge colouring that uses the minimum number of distinct colours. Given a set of packets $P = \{(i_1, o_1), \ldots, (i_m, o_m)\}$, construct from it a multigraph[1] G such that every valid edge colouring of G corresponds to a valid routing $\mathcal{P}$ of P and vice versa. Moreover, the number of colours used by the colouring should equal $|\mathcal{P}|$, which implies that the colouring corresponding to $\mathcal{P}$ is an optimal colouring of G if and only if $\mathcal{P}$ is an optimal routing of the packets in P. Prove that this correspondence between valid/optimal colourings of G and valid/optimal routings of the packets in P holds.

Question 2

Observe that for any valid edge colouring, the edges that are assigned some colour c form a matching, because no two edges incident to any vertex can have the same colour. Thus, we can view the edge colouring problem as the problem of partitioning the edge set of G into matchings. Our goal is to find such a partition consisting of as few matchings as possible. Let Δ be the maximum degree of the vertices in G. Question 3 asks you to prove that every bipartite graph has a matching M that matches all vertices of degree Δ (i.e., all vertices of degree Δ are endpoints of edges in M). The question also asks you to show that such a matching M can be found in $O(nm)$ time. Prove that this implies that a valid edge colouring of G with Δ colours can be found in $O(\Delta nm) \subseteq O\left(n^2 m\right)$ time. Also prove that G does not have any valid edge colouring with fewer than Δ colours, which implies that the colouring produced by your algorithm is an optimal colouring.

[1] A multigraph is like a graph, only it is allowed to contain multiple edges with the same endpoints.

QUESTION 3

Prove that in a bipartite graph, a matching that matches all vertices of maximum degree can be found in $O(nm)$ time.

Hint: *You can use a similar strategy as for the maximum matching problem in bipartite graphs. You start with an empty matching M and then update this matching by repeatedly finding an alternating path P and replacing M with $M \oplus P$. The difference is that you shouldn't necessarily choose P to be an augmenting path, that is, we do not necessarily want $M \oplus P$ to be bigger than M; our goal is to ensure that $M \oplus P$ matches more vertices of degree Δ than M does. To this end, you should prove two things, both under the assumption that G is biparite: (1) As long as M does not match all vertices of degree Δ, there exists an alternating path P in G such that one endpoint of P is an unmatched vertex of degree Δ and either the other endpoint is also unmatched or the path has even length and the other endpoint has degree less than Δ. (2) If such a path exists, it can be found by running alternating BFS from the unmatched vertices of degree Δ in U (or in W if there are no unmatched vertices of degree Δ in U). If you can prove these two things, you should then be able to argue that you can find a matching that matches all degree-Δ vertices.*

Marking Scheme

Question 1 (9 marks)

	Yes	Minor mistakes	Major mistakes	No
Valid edge colouring of the graph corresponds to a valid packet routing	1 mark			0 marks
Valid packet routing corresponds to a valid edge colouring of the graph	1 mark			0 marks
Correct proof that a valid edge colouring corresponds to a valid packet routing	3 marks	2 marks	1 mark	0 marks
Correct proof that a valid packet routing corresponds to a valid edge colouring	3 marks	2 marks	1 mark	0 marks
Correct proof that this implies that valid edge colourings and valid routings correspond to each other	1 mark			0 marks

Question 2 (8 marks marks)

	Yes	Minor mistakes	No
Algorithm computes a colouring with Δ colours	1 marks	0.5 marks	0 marks
Algorithm runs in $O(\Delta nm)$ time	1 marks	0.5 marks	0 marks
Correct proof that the algorithm computes a colouring with Δ colours	2 marks	1 mark	0 marks
Correct proof that the algorithm runs in $O(\Delta nm)$ time	2 marks	1 mark	0 marks
Correct proof that the graph does not have an edge colouring with fewer than Δ colours	2 marks	1 mark	0 marks

Question 3 (12 marks)

	Yes	Minor mistakes	Major mistakes	No
Algorithm computes a matching that matches all vertices of degree Δ	2 marks	1 mark	0 marks	0 marks
Algorithm runs in $O(nm)$ time	2 marks	1 mark	0 marks	0 marks
Correct proof that the algorithm finds a matching that matches all vertices of degree Δ	6 marks	4 marks	2 marks	0 marks
Correct proof that the algorithm runs in $O(nm)$ time	2 marks	1 mark	0 marks	0 marks

Submission Instructions

Follow the submission link for this assignment on the course webpage in the email you should have received from Crowdmark. Upload the assignment as a single PDF file.