**Vlad-Alexandru Gavrila**
**University of Cambridge**

**Solution 1 – 21 points**

We will use the following 3-leveled structure to create the router:

- Level 1: N inputs
- Level 2: K intermediary nodes
- Level 3: N outputs

We group the N inputs evenly into K groups of size N / K. Nodes from each group will be connected with each of the corresponding K nodes from level 2. Each node from level 2 will be connected with each node from level 3.

This grouping produces N + N*K connections. The maximum power will be achieved in the intermediary nodes, and is equal to $N^2$ / K (N output nodes connected with N / K inputs). We can then check for all possible values for K which one produces a solution which satisfies the power and connection constraints.

We find that a suitable value for K is approximately $N^{(1/2)}$. This produces N + $N^{(3/2)}$ connections and $N^{(3/2)}$ maximum power.

**Solution 2 – 47 points**

We will use the following 4-leveled structure to create the router:

- Level 1: N inputs
- Level 2: K intermediary nodes
- Level 3: K intermediary nodes
- Level 4: N outputs

We group the N inputs evenly into K groups of size N / K. Nodes from each of the K groups will be connected with each of the corresponding K nodes from level 2. Similarly, all outputs will be grouped and connected to the corresponding nodes from level 3. Finally, each node from level 2 will be connected with each node from level 3.

This grouping produces 2*N + $K^2$ connections. The maximum power will be achieved in the intermediary nodes, and is equal to $N^2$ / K (N input nodes connected with N / K outputs, or N output nodes connected with N / K inputs). We can then check for all possible values for K which one produces a solution which satisfies the power and connection constraints.

We find that a suitable value for K is approximately $N^{(2/3)}$. This produces 2*N + $N^{(4/3)}$ connections and $N^{(4/3)}$ maximum power.

**Solution 3 – 77 points**

We will use the divide and conquer approach. Let's assume we want to connect the input nodes $\{In_0, \ldots, In_{N-1}\}$ to the output nodes $\{Out_0, \ldots, Out_{N-1}\}$.

If $n = 1$, then we simply add an edge from $In_0$ to $Out_0$ and we are done. Otherwise, we will create a middle layer with nodes $\{NextIn_0, \ldots, NextIn_{N-1}\}$ and we will split this layer into two subsets, $Next_0 = \{NextIn_0, \ldots, NextIn_{[N/2]-1}\}$ and $Next_1 = \{NextIn_{[N/2]}, \ldots, NextIn_{N-1}\}$.

Then, we connect every node from $Next_0$ to the output nodes $\{Out_0, \ldots, Out_{[N/2]-1}\}$ and every node from $Next_1$ to the output nodes $\{Out_{[N/2]}, \ldots, Out_{N-1}\}$ using the same recursive approach. Now, it is sufficient to connect every input node $\{In_0, \ldots, In_{N-1}\}$ to at least one node from $Next_0$ and at least one node from $Next_1$ in order to assure that every input node is connected to every output node.

To do this, we will connect $In_i$ to $NextIn_i$ and $NextIn_{N/2+(i \bmod (N-[N/2]))}$ for $0 \le i < N/2$ and we will connect $In_i$ to $NextIn_i$ and $NextIn_{(i-[N/2]) \bmod [N/2]}$ for $N/2 \le i < n$.

Let $E(n)$ be the number of edges used to connect $n$ input nodes to $n$ output nodes. The recurrence for this number is $E(1) = 1$, $E(N) = E([N/2]) + E(N-[N/2]) + 2N$. We can solve this even programmatically and we find $E(N) = 2N * \log_2 N + N$.

Let us analyze the maximum cost of a node if we connect $n$ input nodes to $n$ output nodes. Every input node and every output node will have a cost of exactly $N$. Every node at the $k$-th intermediate layer will be connected to $N/2^k$ output nodes and there will be less than $2^{k+1}$ input nodes connected to it. Thus, the maximum cost is less than $2N$.

**Solution 4 – 100 points**

We will optimize the previous solution, in order the balance the number of edges with the maximum cost of a node.

If we try to connect $N$ input nodes to $N$ output nodes, we will create $[N/t]$ input buckets and $[N/t]$ output buckets. We will create an intermediate node for every bucket and we will connect every node $i$ to the $[i/t]$-th corresponding intermediate node ($0 \le i < n$). Thus, we have two middle input and output layers with $[N/t]$ nodes each, which we will connect using the divide and conquer approach.

This solution decreases the number of edges with a factor comparable to $t$ and increases the maximum cost of a node with a factor comparable to $t^2$. The optimal value of $t$ can be determined using various techniques, even through manual experimentation. A reasonable value for this threshold is $t = 4$.