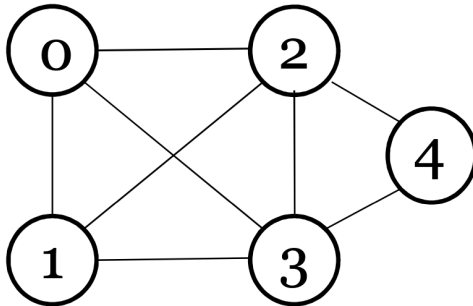


IN3200/IN4200, Spring 2021
Comments about home exam 1

Connectivity graph

- Nodes represent individual data objects
- Edges represent **direct-connections** between pairs of “nearest neighbors”
- Symmetry: if node u is a nearest neighbor of node v , then v is a nearest neighbor of u



Data storage format 1 for connectivity graph

A 2D table of values “0” and “1” (implemented as a 2D array of type `char**`)

row 0	0	1	1	1	0
row 1	1	0	1	1	0
row 2	1	1	0	1	1
row 3	1	1	1	0	1
row 4	0	0	1	1	0

Note: The number of values of 1 in the 2D table is twice the number of edges in the connectivity graph

Data storage format 2 for connectivity graph

Compressed row storage (CRS) implemented as two arrays of integer values (see Section 3.6.1 in textbook)

row_ptr: 0, 3, 6, 10, 14, 16

col_idx: 1, 2, 3, 0, 2, 3, 0, 1, 3, 4, 0, 1, 2, 4, 2, 3

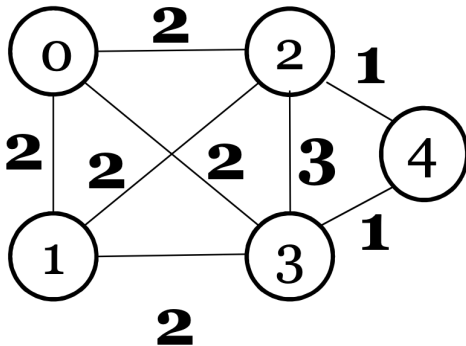
Length of row_ptr: number of nodes in the connectivity graph + 1

Length of col_idx: twice the number of edges

SNN graph

One measure of “similarity” between two directly connected nodes u and v is the number of their shared nearest neighbors (SNNs)

How many other nodes are directly connected with both u and v ?



Data storage formats for SNN graph

Either as a 2D table of type `int**`

Or by adding an additional integer array `SNN_val` to the arrays of `row_ptr` and `col_idx` of the CRS format (same length for `SNN_val` and `col_idx`)

Clustering based on SNN (only relevant for IN4200 students)

Given an SNN graph and a threshold value τ , we want to find “clusters”.

Each cluster is a subset of the nodes, where each node in the cluster is directly connected with *at least* another node in the same cluster, with the number of SNNs between them being equal or larger than the threshold value τ .

For example, nodes 0,1,2,3 in the example SNN graph will form a cluster for $\tau = 2$.

C programming

- Two functions for constructing a connectivity graph based on info stored in a given data file
- Two functions for creating a SNN graph from a given connectivity graph
 - OpenMP parallelization of these two functions
- **Only for IN4200 students:** One function for identifying all the nodes inside an SNN graph that belong to the same cluster with a given node
- One test (`main`) program that makes use of the above functions

A short note (in PDF format) that explains the basic idea (and algorithm) behind each function, as well as efficiency considerations (if relevant)

Hint: How to construct a 1D array inside a C function?

Challenge: We want the constructed array to “live on” after the function call

An illustrating example:

```
void array_construction (int **array1D, int length) {  
    int i;  
    *array1D = (int*)malloc(length*sizeof(int));  
    for (i=0; i<length; i++)  
        (*array1D)[i] = i;  
}
```

To make use of the function:

```
int *a;  
array_construction (&a, 100);  
// ...  
free (a);
```