

Algorithms and Datastructures (HS2024)

Week 8

Rui Zhang

November 11, 2024

Contents

1	Revision Theory	1
1.1	Undirected Graphs	1
1.2	Directed Graphs	2
1.3	Graph Properties & Terminology	2
1.4	Graph Structures	3
1.5	Things we do not talk about	3
2	Exercises	3
2.1	Exercise Sheet 8 - Priority List	3
2.2	Exercise Sheet 6 - Feedback	3
3	Supplementary Exercises	4
3.1	Euler	4
3.2	Short Questions About Graphs - Legacy Version	4
	Shoutouts to Arwed Walke for the title for this weeks website entry.	

1 Revision Theory

1.1 Undirected Graphs

An undirected Graph is a tuple $G = (V, E)$, where V and E are sets. V is the set of vertices / nodes. E is the set of edges

Just like when we talked about trees, nodes / vertices are the round circles and the edges are the lines connecting two vertices. Usually, we use the notation $|V| = n$, $|E| = m$. We denote that an edge exists between two nodes $u, v \in V$ by writing $\{u, v\} \in E$. So note that the elements of E are the edges which are represented as sets of two nodes. This is sort of intuitive because the order of the nodes is irrelevant since we have an undirected graph. Lastly, it should be mentioned that $\deg(v)$ is the number of neighbors of the node v .

1.2 Directed Graphs

A directed Graph is also the same tuple $G = (V, E)$, but this time, the set of edges E is not a set of sets, but a set of tuples (u, v) , where for $\{u, v\} \in E$, $(u, v) \in E$ if and only if the **directed** edge going from u to v exists in E .

Here, we also differentiate between outgoing and ingoing edges and thus in-degree and out-degree.

$\text{deg}^+(v) :=$ out degree, number of outgoing edges

$\text{deg}^-(v) :=$ in degree, number of ingoing edges

1.3 Graph Properties & Terminology

We say that two vertices $u, v \in V$ are adjacent, if and only if there exists an edge $(u, v) \in E$ in the directed case and $\{u, v\} \in E$ in the undirected case. We say that an edge $e \in E$ is incident to a node $u \in V$, if and only if e is of the form $e = (u, v)$, $e = (v, u)$ if the graph is directed and else $e = \{v, u\}$.

For undirected graphs, we have that

$$\sum_{v \in V} \text{deg}(v) = 2|E|$$

This stems from the intuition that by counting all neighbors of every node, we counted every edge in the graph twice.

For directed graphs, this is equivalent to:

$$\sum_{v \in V} \text{deg}^+(v) + \text{deg}^-(v) = 2|E|$$

These equations are called Handshake Lemma.

A walk is a sequence of nodes such that the edges existing in the graph allow for this route. Formally, a walk W is

$$W = (v_0, v_1, v_2, \dots, v_{l-1}, v_l) \text{ for some length } l$$

such that $(v_i, v_{i+1}) \in E$ for all $i \in 1, 2, 3, \dots, l-1$

A path P is also a sequence of nodes (so it is a walk), but no vertices are repeated.

A closed walk C is a walk that starts and ends at the same node: $v_0 = v_l$ and such that the length of the walk is at least 2: $l \geq 2$

A cycle K is a path which starts and ends at the same node and $l \geq 3$.

We say that u reaches v if there exists a walk starting at u and ending in v .

We say that a graph is connected if for every pair $u, v \in V$, $u \neq v$ u reaches v .

The connected components of a graph are the equivalence classes of the "reaches" relationship.

1.4 Graph Structures

A eulerian walk is a walk in which every edge of the graph is used **exactly** once. A eulerian walk exists if the number of nodes of odd degree is less than or equal to 2.

A eulerian closed walk is a closed walk in which every edge of the graph is used **exactly** once. A eulerian closed walk exists if every node has even degree.

A hamiltonian path is a path which goes through every node of the graph **exactly** once.

A hamiltonian cycle is a cycle which goes through every node of the graph **exactly** once.

Via the "walk" algorithm you have seen in the lecture you are able to find a eulerian cycle in a graph in $O(m + n)$

1.5 Things we do not talk about

There are things we do not / rarely talk about in this course, like the unemployed uncle who shows up to family events only to talk about politics.

One of them are self loops: $(v, v) \in E$ or $\{v, v\} \in E$. the other are multi-graphs, which are graphs which contain some edges multiple times.

2 Exercises

2.1 Exercise Sheet 8 - Priority List

1. 8.1), 8.5) are the most important exercises of this sheet 8.1) develops essential skills in Graph modelling and 8.5) is very likely exam relevant
2. 8.3) teaches you the way to think when developing graph algorithms.
3. 8.4) teaches you the important connection between trees and graphs.
4. 8.2) is a funny little exercise in graph understanding

2.2 Exercise Sheet 6 - Feedback

1. In the DP exercise, a lot of you had written the recursive relationship between one DP entry to other previous ones in the "DP-Table Definition" Section of the exercise. The definition is usually an intuitive explanation of one entry in the table in words. In this case, it was simply " $DP[i] := A_i$ ".
2. In the same exercise, in subtasks b), when calculating the C for the lower bound of the runtime, many of you had correctly identified one such recursive relationship, namely $T(n) = T(n - 1) + T(n - 2) + d$ for some constant d . However, this only holds for the case where n is odd. Additionally, many of you referred to a past exercise, but in an exam scenario, be sure to expect to have to do this exercise via induction.

3. In the exercise maximum almost subarray sum, many had not actually calculated values for R' and S and Instead had written code. This is good practice. However, in the future, if the task says to compute, then it means to calculate it by hand and if it says to write an algorithm, only then switch to pseudocode.

3 Supplementary Exercises

3.1 Euler

Explain the intuition behind the necessary conditions for the existence of eulerian walks and closed eulerian walks.

3.2 Short Questions About Graphs - Legacy Version

This exercise is from last years sheet for the same week.

In the following, assume that $G = (V, E)$ is an undirected graph, $n := |V|$ and $m := |E|$ Either prove the statement or show a counterexample.

- Let $v \neq w \in V$. Prove that if there exists a walk with endpoints v and w , then there is a path with endpoints v and w .
- Every graph with $m \geq n$ is connected
- If G contains a Hamiltonian path, then G contains a Eulerian walk.
- If every vertex of a non-empty graph G has degree at least 2, then G contains a cycle
- Suppose in a graph G every pair of vertices v, w has a common neighbor (i.e., for all distinct vertices v, w , there is a vertex x such that $\{v, x\}$ and $\{w, x\}$ are both edges). Then there exists a vertex p in G which is a neighbour of every other vertex in G (i.e., p has degree $n - 1$).