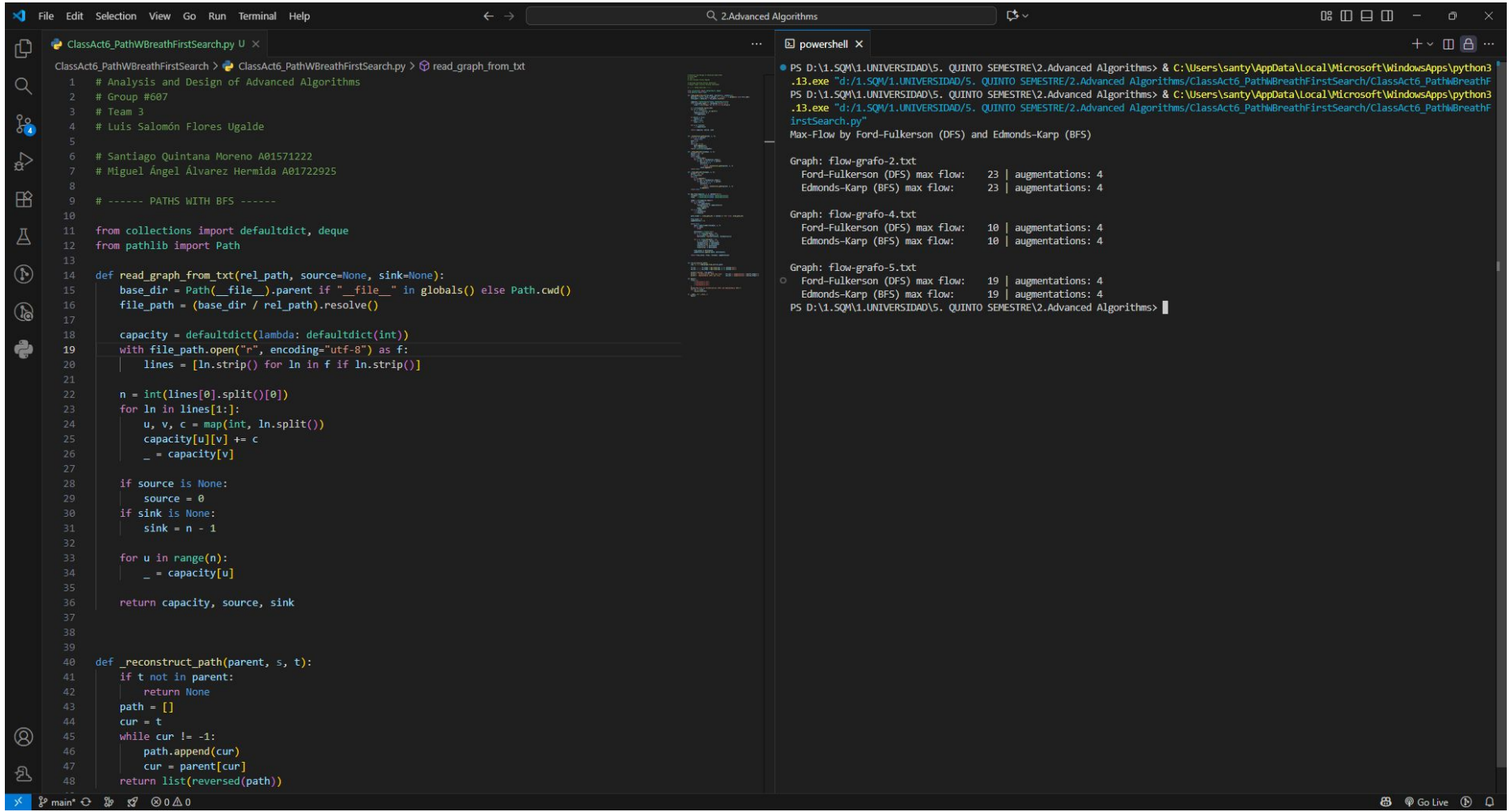
Tecnológico de Monterrey

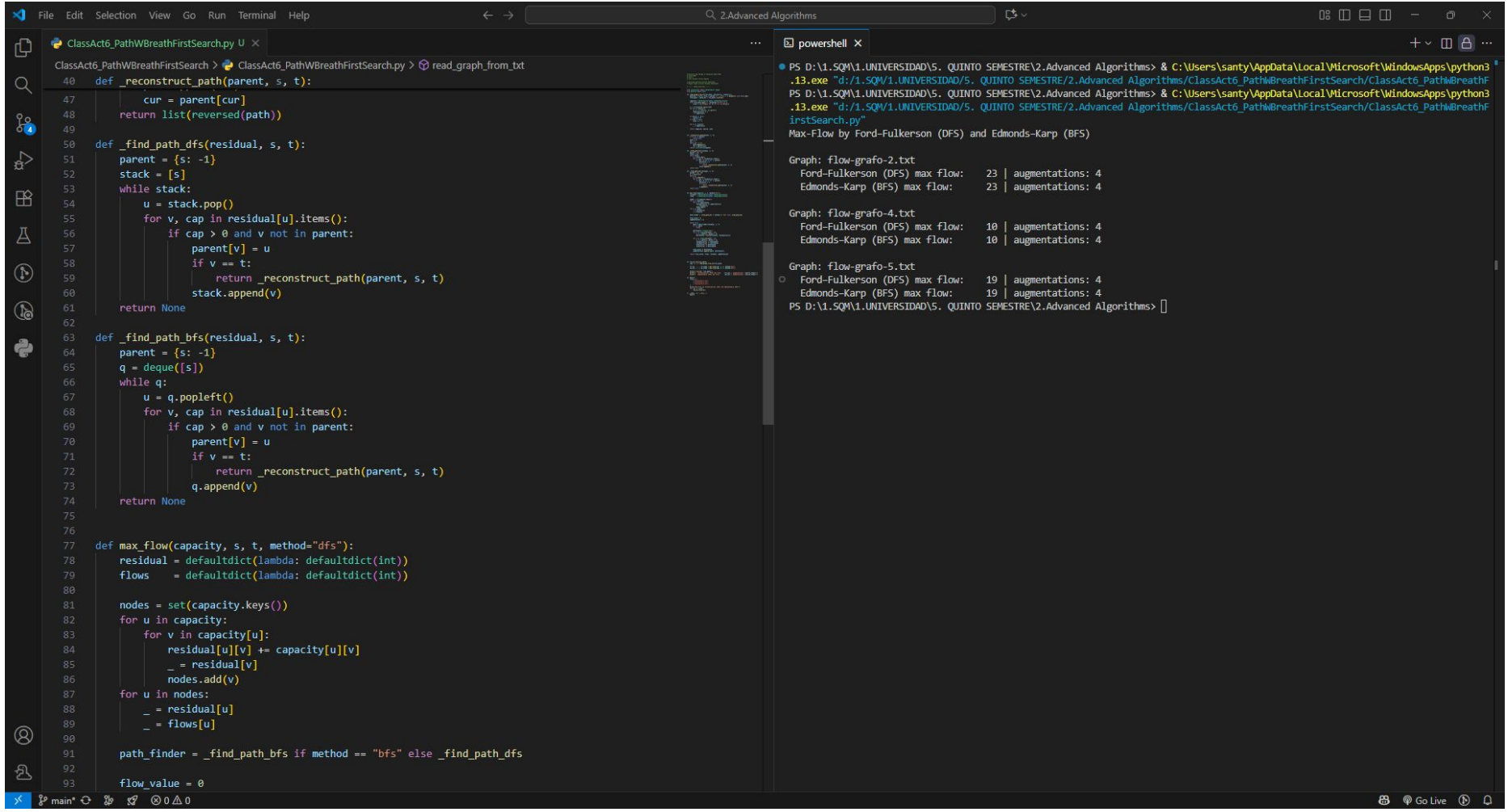
Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Engineering in Computational Technologies
Analysis and Design of Advanced Algorithms

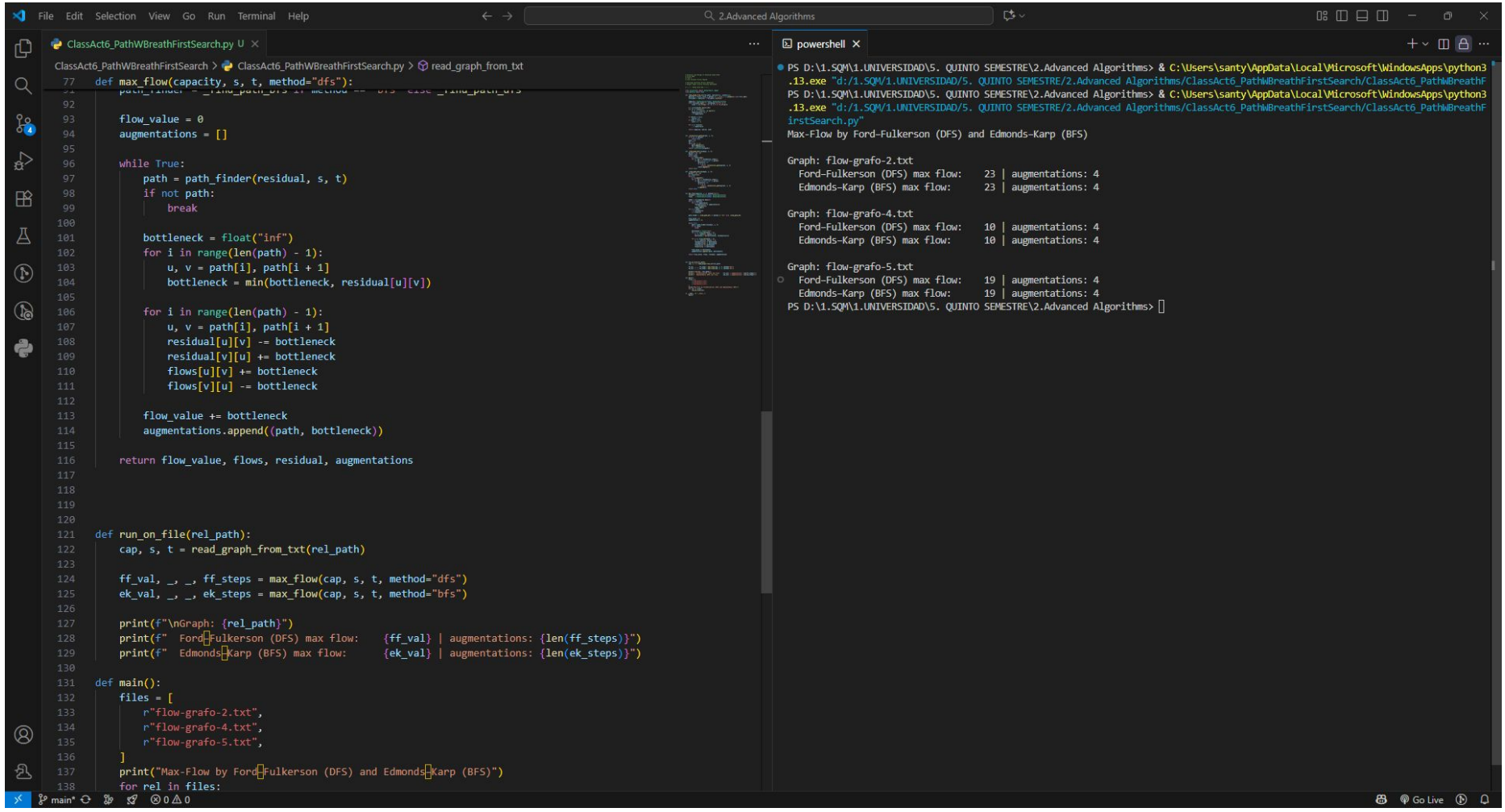
Class Activity 6:
Paths with Breadth First Search

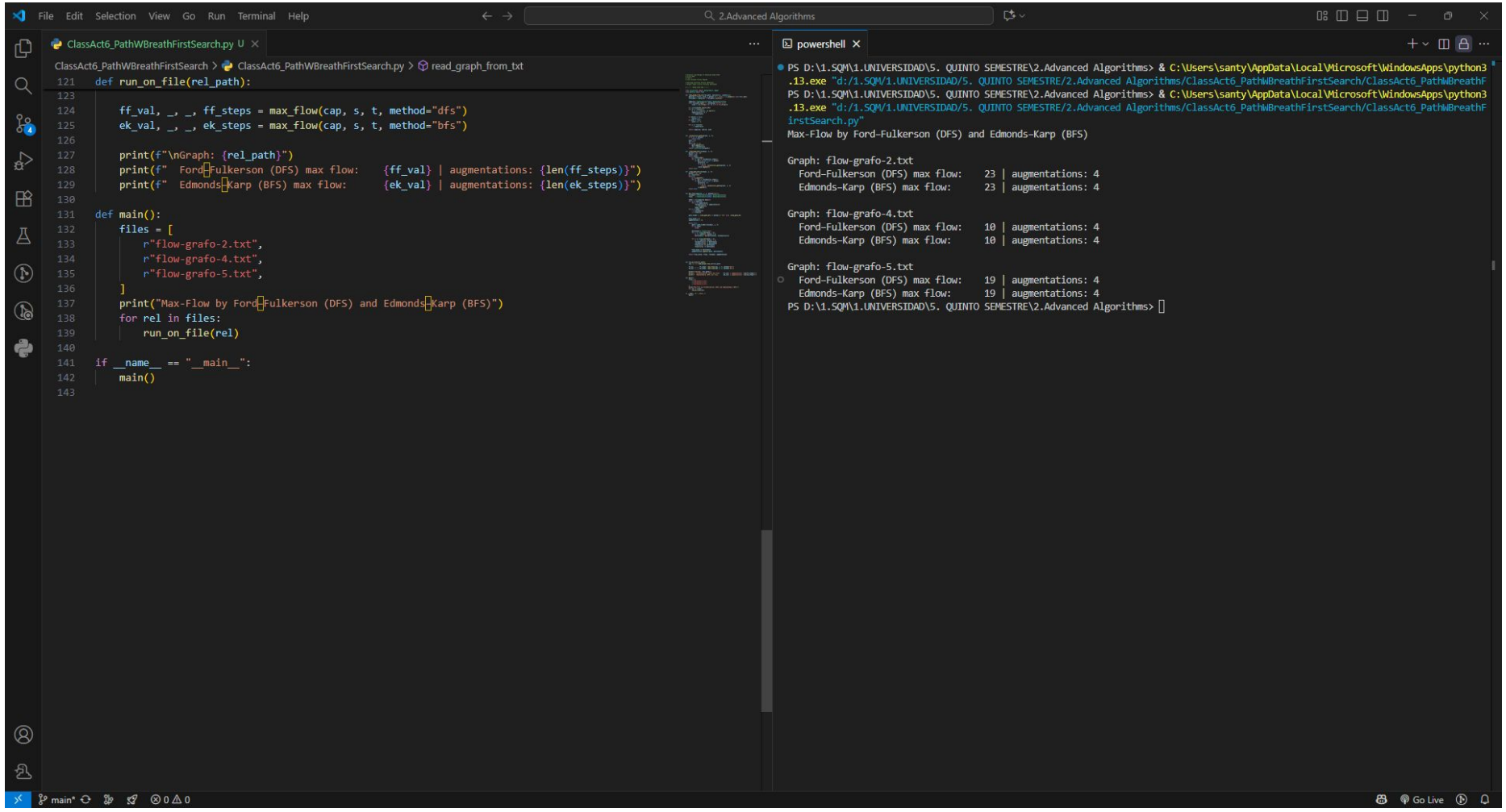
Group: 607
Team #3
Luis Salomón Flores Ugalde

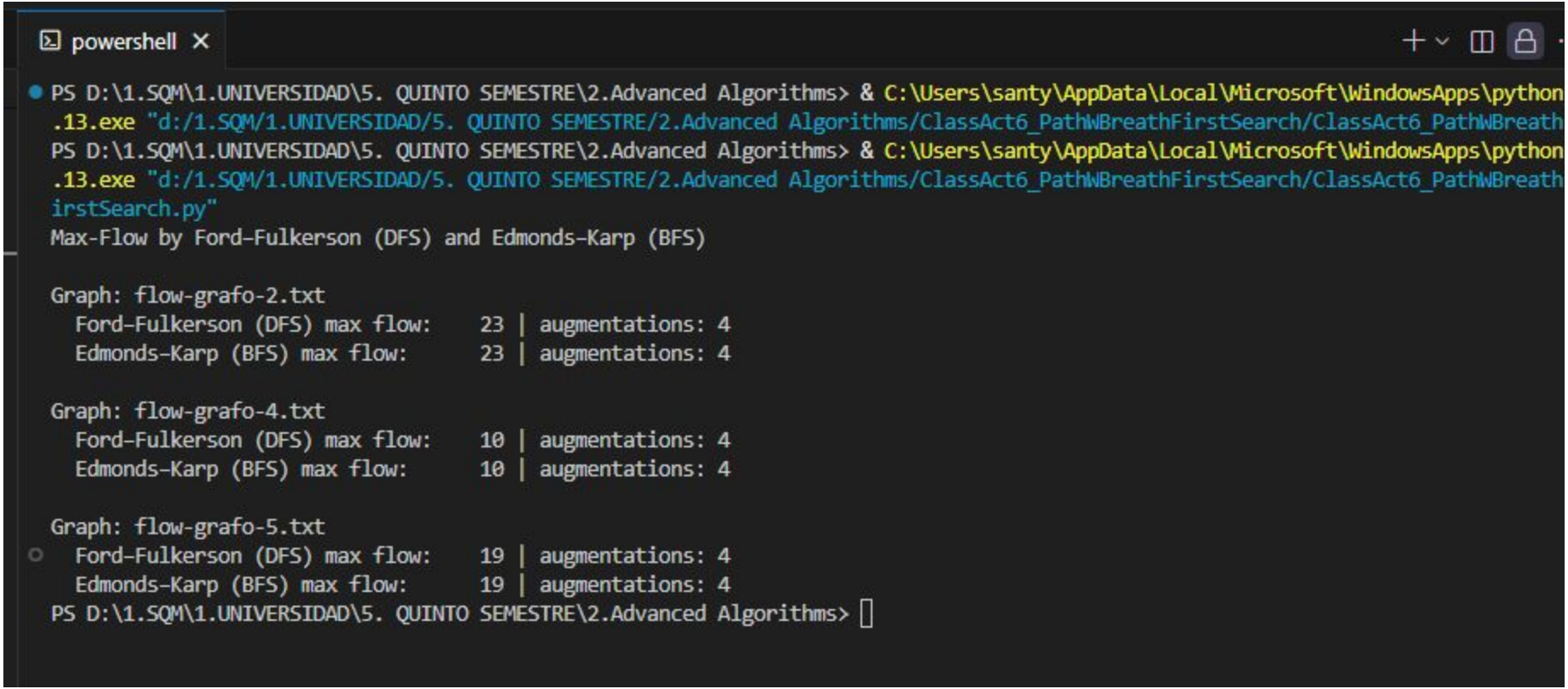
Santiago Quintana Moreno A01571222
Miguel Ángel Álvarez Hermida a01722925


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct6_PathWBreathFirstSearch.py U
ClassAct6_PathWBreathFirstSearch > ClassAct6_PathWBreathFirstSearch.py > read_graph_from_txt
# Analysis and Design of Advanced Algorithms
# Group #607
# Team 3
# Luis Salomón Flores Ugalde
# Santiago Quintana Moreno A01571222
# Miguel Ángel Álvarez Hermida A01722925
# ------ PATHS WITH BFS ------
from collections import defaultdict, deque
from pathlib import Path
def read_graph_from_txt(rel_path, source=None, sink=None):
base_dir = Path(__file__).parent if "__file__" in globals() else Path.cwd()
file_path = (base_dir / rel_path).resolve()
capacity = defaultdict(lambda: defaultdict(int))
with file_path.open("r", encoding="utf-8") as f:
lines = [ln.strip() for ln in f if ln.strip()]
n = int(lines[0].split()[0])
for ln in lines[1:]:
u, v, c = map(int, ln.split())
capacity[u][v] += c
_ = capacity[v]
if source is None:
source = 0
if sink is None:
sink = n - 1
for u in range(n):
_ = capacity[u]
return capacity, source, sink
def _reconstruct_path(parent, s, t):
if t not in parent:
return None
path = []
cur = t
while cur != -1:
path.append(cur)
cur = parent[cur]
return list(reversed(path))
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreathFirstSearch.py"
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreathFirstSearch.py"
Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)
Graph: flow-grafo-2.txt
Ford-Fulkerson (DFS) max flow: 23 | augmentations: 4
Edmonds-Karp (BFS) max flow: 23 | augmentations: 4
Graph: flow-grafo-4.txt
Ford-Fulkerson (DFS) max flow: 10 | augmentations: 4
Edmonds-Karp (BFS) max flow: 10 | augmentations: 4
Graph: flow-grafo-5.txt
Ford-Fulkerson (DFS) max flow: 19 | augmentations: 4
Edmonds-Karp (BFS) max flow: 19 | augmentations: 4
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct6_PathWBreathFirstSearch.py U
ClassAct6_PathWBreathFirstSearch > ClassAct6_PathWBreathFirstSearch.py > read_graph_from_txt
def _reconstruct_path(parent, s, t):
cur = parent[cur]
return list(reversed(path))
def _find_path_dfs(residual, s, t):
parent = {s: -1}
stack = [s]
while stack:
u = stack.pop()
for v, cap in residual[u].items():
if cap > 0 and v not in parent:
parent[v] = u
if v == t:
return _reconstruct_path(parent, s, t)
stack.append(v)
return None
def _find_path_bfs(residual, s, t):
parent = {s: -1}
q = deque([s])
while q:
u = q.popleft()
for v, cap in residual[u].items():
if cap > 0 and v not in parent:
parent[v] = u
if v == t:
return _reconstruct_path(parent, s, t)
q.append(v)
return None
def max_flow(capacity, s, t, method="dfs"):
residual = defaultdict(lambda: defaultdict(int))
flows = defaultdict(lambda: defaultdict(int))
nodes = set(capacity.keys())
for u in capacity:
for v in capacity[u]:
residual[u][v] += capacity[u][v]
_ = residual[v]
nodes.add(v)
for u in nodes:
_ = residual[u]
_ = flows[u]
path_finder = _find_path_bfs if method == "bfs" else _find_path_dfs
flow_value = 0
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreathFirstSearch.py"
Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)
Graph: flow-grafo-2.txt
Ford-Fulkerson (DFS) max flow: 23 | augmentations: 4
Edmonds-Karp (BFS) max flow: 23 | augmentations: 4
Graph: flow-grafo-4.txt
Ford-Fulkerson (DFS) max flow: 10 | augmentations: 4
Edmonds-Karp (BFS) max flow: 10 | augmentations: 4
Graph: flow-grafo-5.txt
Ford-Fulkerson (DFS) max flow: 19 | augmentations: 4
Edmonds-Karp (BFS) max flow: 19 | augmentations: 4
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct6_PathWBreathFirstSearch.py U
ClassAct6_PathWBreathFirstSearch > ClassAct6_PathWBreathFirstSearch.py > read_graph_from_txt
def max_flow(capacity, s, t, method="dfs"):
flow_value = 0
augmentations = []
while True:
path = path_finder(residual, s, t)
if not path:
break
bottleneck = float("inf")
for i in range(len(path) - 1):
u, v = path[i], path[i + 1]
bottleneck = min(bottleneck, residual[u][v])
for i in range(len(path) - 1):
u, v = path[i], path[i + 1]
residual[u][v] -= bottleneck
residual[v][u] += bottleneck
flows[u][v] += bottleneck
flows[v][u] -= bottleneck
flow_value += bottleneck
augmentations.append((path, bottleneck))
return flow_value, flows, residual, augmentations
def run_on_file(rel_path):
cap, s, t = read_graph_from_txt(rel_path)
ff_val, _, _, ff_steps = max_flow(cap, s, t, method="dfs")
ek_val, _, _, ek_steps = max_flow(cap, s, t, method="bfs")
print(f"\nGraph: {rel_path}")
print(f" Ford-Fulkerson (DFS) max flow: {ff_val} | augmentations: {len(ff_steps)}")
print(f" Edmonds-Karp (BFS) max flow: {ek_val} | augmentations: {len(ek_steps)}")
def main():
files = [
r"flow-grafo-2.txt",
r"flow-grafo-4.txt",
r"flow-grafo-5.txt",
]
print("Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)")
for rel in files:
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreathFirstSearch.py"
Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)
Graph: flow-grafo-2.txt
Ford-Fulkerson (DFS) max flow: 23 | augmentations: 4
Edmonds-Karp (BFS) max flow: 23 | augmentations: 4
Graph: flow-grafo-4.txt
Ford-Fulkerson (DFS) max flow: 10 | augmentations: 4
Edmonds-Karp (BFS) max flow: 10 | augmentations: 4
Graph: flow-grafo-5.txt
Ford-Fulkerson (DFS) max flow: 19 | augmentations: 4
Edmonds-Karp (BFS) max flow: 19 | augmentations: 4
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct6_PathWBreathFirstSearch.py U
ClassAct6_PathWBreathFirstSearch > ClassAct6_PathWBreathFirstSearch.py > read_graph_from_txt
def run_on_file(rel_path):
ff_val, _, _, ff_steps = max_flow(cap, s, t, method="dfs")
ek_val, _, _, ek_steps = max_flow(cap, s, t, method="bfs")
print(f"\nGraph: {rel_path}")
print(f" Ford-Fulkerson (DFS) max flow: {ff_val} | augmentations: {len(ff_steps)}")
print(f" Edmonds-Karp (BFS) max flow: {ek_val} | augmentations: {len(ek_steps)}")
def main():
files = [
r"flow-grafo-2.txt",
r"flow-grafo-4.txt",
r"flow-grafo-5.txt",
]
print("Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)")
for rel in files:
run_on_file(rel)
if __name__ == "__main__":
main()
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreathFirstSearch.py"
Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)
Graph: flow-grafo-2.txt
Ford-Fulkerson (DFS) max flow: 23 | augmentations: 4
Edmonds-Karp (BFS) max flow: 23 | augmentations: 4
Graph: flow-grafo-4.txt
Ford-Fulkerson (DFS) max flow: 10 | augmentations: 4
Edmonds-Karp (BFS) max flow: 10 | augmentations: 4
Graph: flow-grafo-5.txt
Ford-Fulkerson (DFS) max flow: 19 | augmentations: 4
Edmonds-Karp (BFS) max flow: 19 | augmentations: 4
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
Go Live


powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreath
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/ClassAct6_PathWBreathFirstSearch/ClassAct6_PathWBreathirstSearch.py"
Max-Flow by Ford-Fulkerson (DFS) and Edmonds-Karp (BFS)
Graph: flow-grafo-2.txt
Ford-Fulkerson (DFS) max flow: 23 | augmentations: 4
Edmonds-Karp (BFS) max flow: 23 | augmentations: 4
Graph: flow-grafo-4.txt
Ford-Fulkerson (DFS) max flow: 10 | augmentations: 4
Edmonds-Karp (BFS) max flow: 10 | augmentations: 4
Graph: flow-grafo-5.txt
Ford-Fulkerson (DFS) max flow: 19 | augmentations: 4
Edmonds-Karp (BFS) max flow: 19 | augmentations: 4
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>

https://colab.research.google.com/drive/1_Ha60Mu4n2Rv73LMrAdA1LFSTfroMsuI?usp=sharing