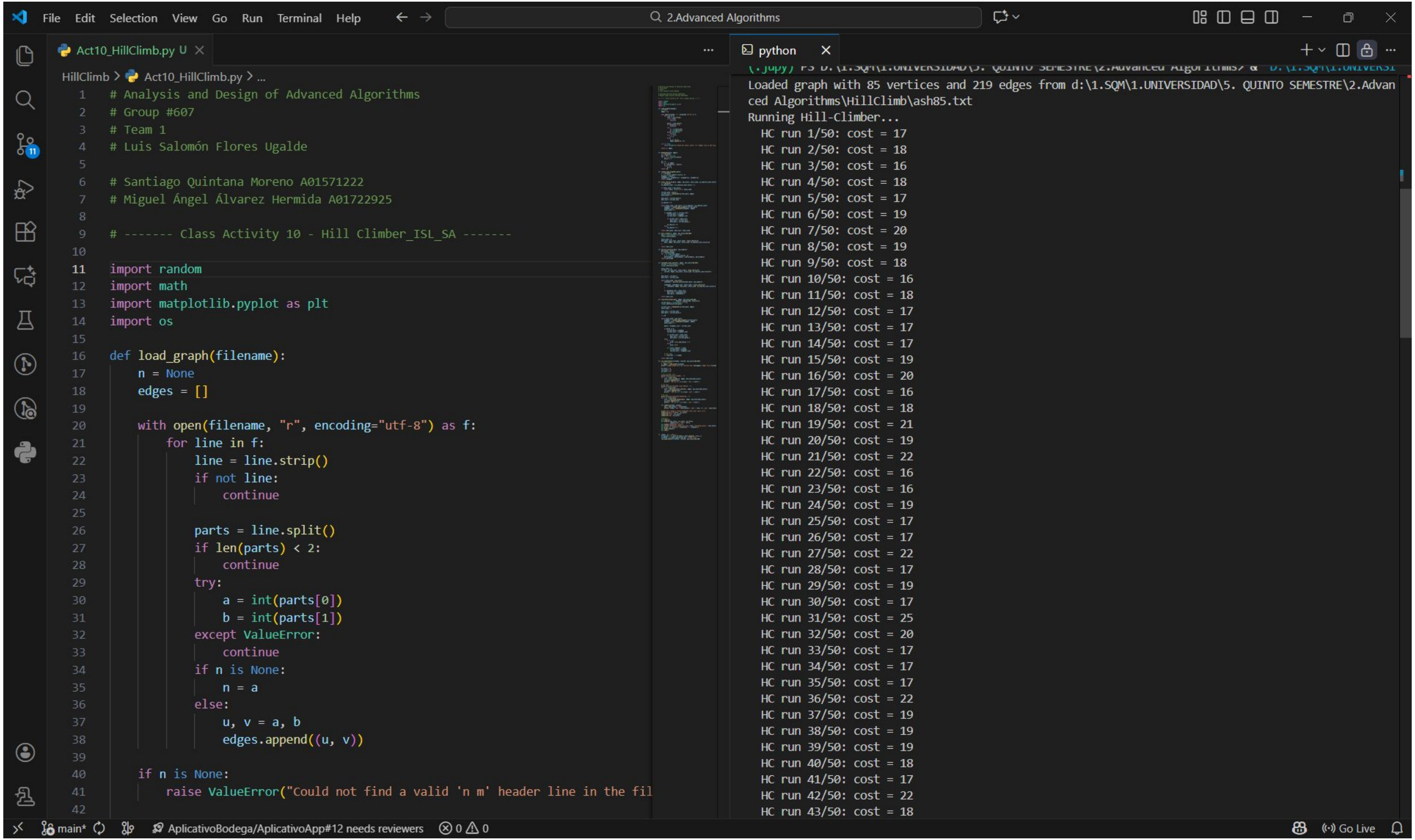
Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Engineering in Computational Technologies
Analysis and Design of Advanced Algorithms

Class Activity 10:
Hill Climb

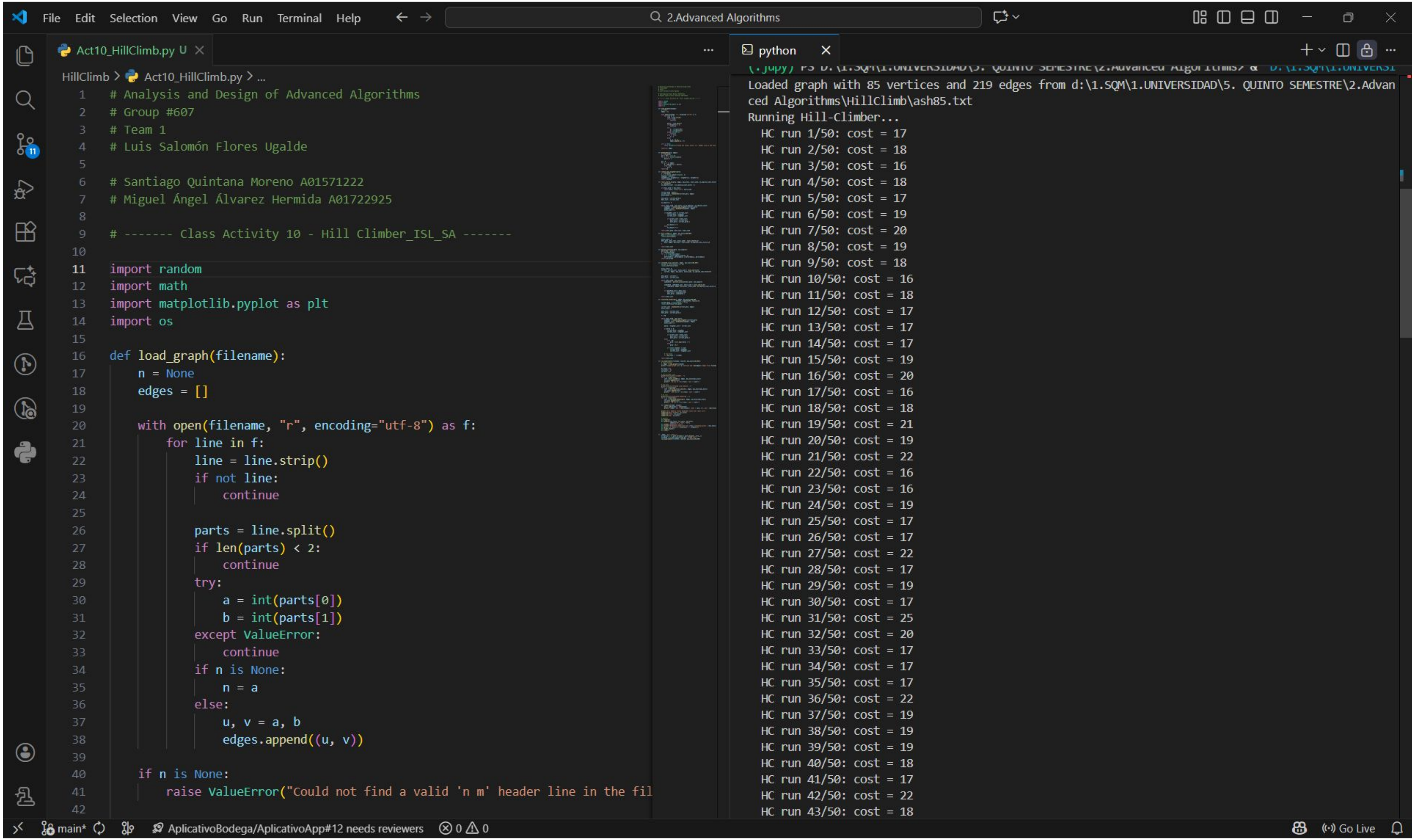
Group: 607
Team #3
Luis Salomón Flores Ugalde

Santiago Quintana Moreno A01571222
Miguel Ángel Álvarez Hermida a01722925



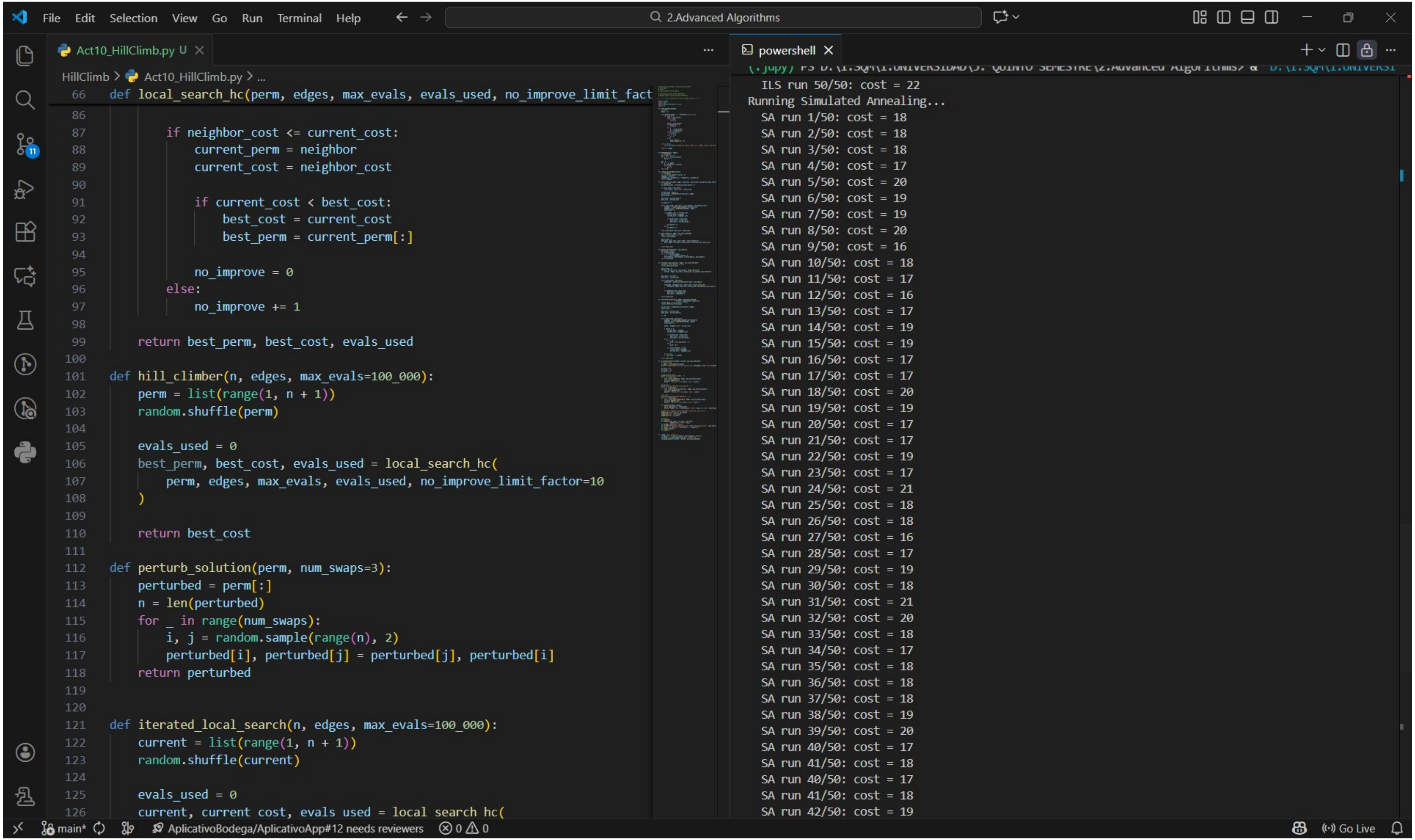
```python
# Analysis and Design of Advanced Algorithms
# Group #607
# Team 1
# Luis Salomón Flores Ugalde

# Santiago Quintana Moreno A01571222
# Miguel Ángel Álvarez Hermida A01722925

# ------- Class Activity 10 - Hill Climber_ISL_SA -------

import random
import math
import matplotlib.pyplot as plt
import os

def load_graph(filename):
    n = None
    edges = []

    with open(filename, "r", encoding="utf-8") as f:
        for line in f:
            line = line.strip()
            if not line:
                continue

            parts = line.split()
            if len(parts) < 2:
                continue
            try:
                a = int(parts[0])
                b = int(parts[1])
            except ValueError:
                continue
            if n is None:
                n = a
            else:
                u, v = a, b
                edges.append((u, v))

    if n is None:
        raise ValueError("Could not find a valid 'n m' header line in the fil

```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
Loaded graph with 85 vertices and 219 edges from d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advan
ced Algorithms\HillClimb\ash85.txt
Running Hill-Climber...
  HC run 1/50: cost = 17
  HC run 2/50: cost = 18
  HC run 3/50: cost = 16
  HC run 4/50: cost = 18
  HC run 5/50: cost = 17
  HC run 6/50: cost = 19
  HC run 7/50: cost = 20
  HC run 8/50: cost = 19
  HC run 9/50: cost = 18
  HC run 10/50: cost = 16
  HC run 11/50: cost = 18
  HC run 12/50: cost = 17
  HC run 13/50: cost = 17
  HC run 14/50: cost = 17
  HC run 15/50: cost = 19
  HC run 16/50: cost = 20
  HC run 17/50: cost = 16
  HC run 18/50: cost = 18
  HC run 19/50: cost = 21
  HC run 20/50: cost = 19
  HC run 21/50: cost = 22
  HC run 22/50: cost = 16
  HC run 23/50: cost = 16
  HC run 24/50: cost = 19
  HC run 25/50: cost = 17
  HC run 26/50: cost = 17
  HC run 27/50: cost = 22
  HC run 28/50: cost = 17
  HC run 29/50: cost = 19
  HC run 30/50: cost = 17
  HC run 31/50: cost = 25
  HC run 32/50: cost = 20
  HC run 33/50: cost = 17
  HC run 34/50: cost = 17
  HC run 35/50: cost = 17
  HC run 36/50: cost = 22
  HC run 37/50: cost = 19
  HC run 38/50: cost = 19
  HC run 39/50: cost = 19
  HC run 40/50: cost = 18
  HC run 41/50: cost = 17
  HC run 42/50: cost = 22
  HC run 43/50: cost = 18
```


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Act10_HillClimb.py U
HillClimb > Act10_HillClimb.py > ...
python
main*
AplicativoBodega/AplicativoApp#12 needs reviewers
⊗ 0 ⚠ 0
Go Live

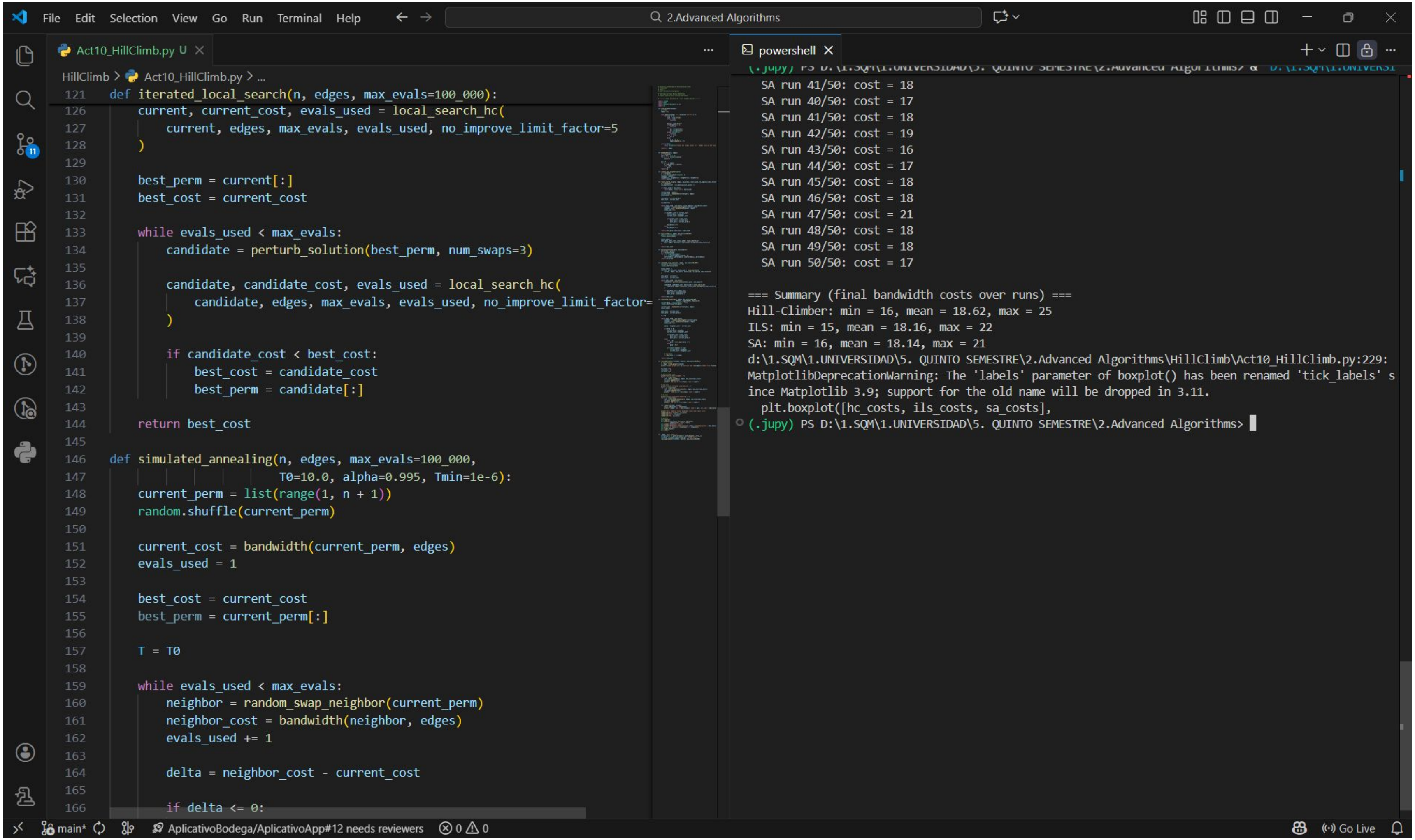
```python
# Analysis and Design of Advanced Algorithms
# Group #607
# Team 1
# Luis Salomón Flores Ugalde

# Santiago Quintana Moreno A01571222
# Miguel Ángel Álvarez Hermida A01722925

# ------- Class Activity 10 - Hill Climber_ISL_SA -------

import random
import math
import matplotlib.pyplot as plt
import os

def load_graph(filename):
    n = None
    edges = []

    with open(filename, "r", encoding="utf-8") as f:
        for line in f:
            line = line.strip()
            if not line:
                continue

            parts = line.split()
            if len(parts) < 2:
                continue
            try:
                a = int(parts[0])
                b = int(parts[1])
            except ValueError:
                continue
            if n is None:
                n = a
            else:
                u, v = a, b
                edges.append((u, v))

    if n is None:
        raise ValueError("Could not find a valid 'n m' header line in the fil
```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
Loaded graph with 85 vertices and 219 edges from d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advan
ced Algorithms\HillClimb\ash85.txt
Running Hill-Climber...
  HC run 1/50: cost = 17
  HC run 2/50: cost = 18
  HC run 3/50: cost = 16
  HC run 4/50: cost = 18
  HC run 5/50: cost = 17
  HC run 6/50: cost = 19
  HC run 7/50: cost = 20
  HC run 8/50: cost = 19
  HC run 9/50: cost = 18
  HC run 10/50: cost = 16
  HC run 11/50: cost = 18
  HC run 12/50: cost = 17
  HC run 13/50: cost = 17
  HC run 14/50: cost = 17
  HC run 15/50: cost = 19
  HC run 16/50: cost = 20
  HC run 17/50: cost = 16
  HC run 18/50: cost = 18
  HC run 19/50: cost = 21
  HC run 20/50: cost = 19
  HC run 21/50: cost = 22
  HC run 22/50: cost = 16
  HC run 23/50: cost = 16
  HC run 24/50: cost = 19
  HC run 25/50: cost = 17
  HC run 26/50: cost = 17
  HC run 27/50: cost = 22
  HC run 28/50: cost = 17
  HC run 29/50: cost = 19
  HC run 30/50: cost = 17
  HC run 31/50: cost = 25
  HC run 32/50: cost = 20
  HC run 33/50: cost = 17
  HC run 34/50: cost = 17
  HC run 35/50: cost = 17
  HC run 36/50: cost = 22
  HC run 37/50: cost = 19
  HC run 38/50: cost = 19
  HC run 39/50: cost = 19
  HC run 40/50: cost = 18
  HC run 41/50: cost = 17
  HC run 42/50: cost = 22
  HC run 43/50: cost = 18
```


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Act10_HillClimb.py U
HillClimb > Act10_HillClimb.py > ...
powershell
main*
AplicativoBodega/AplicativoApp#12 needs reviewers
⊗ 0 ⚠ 0
Go Live

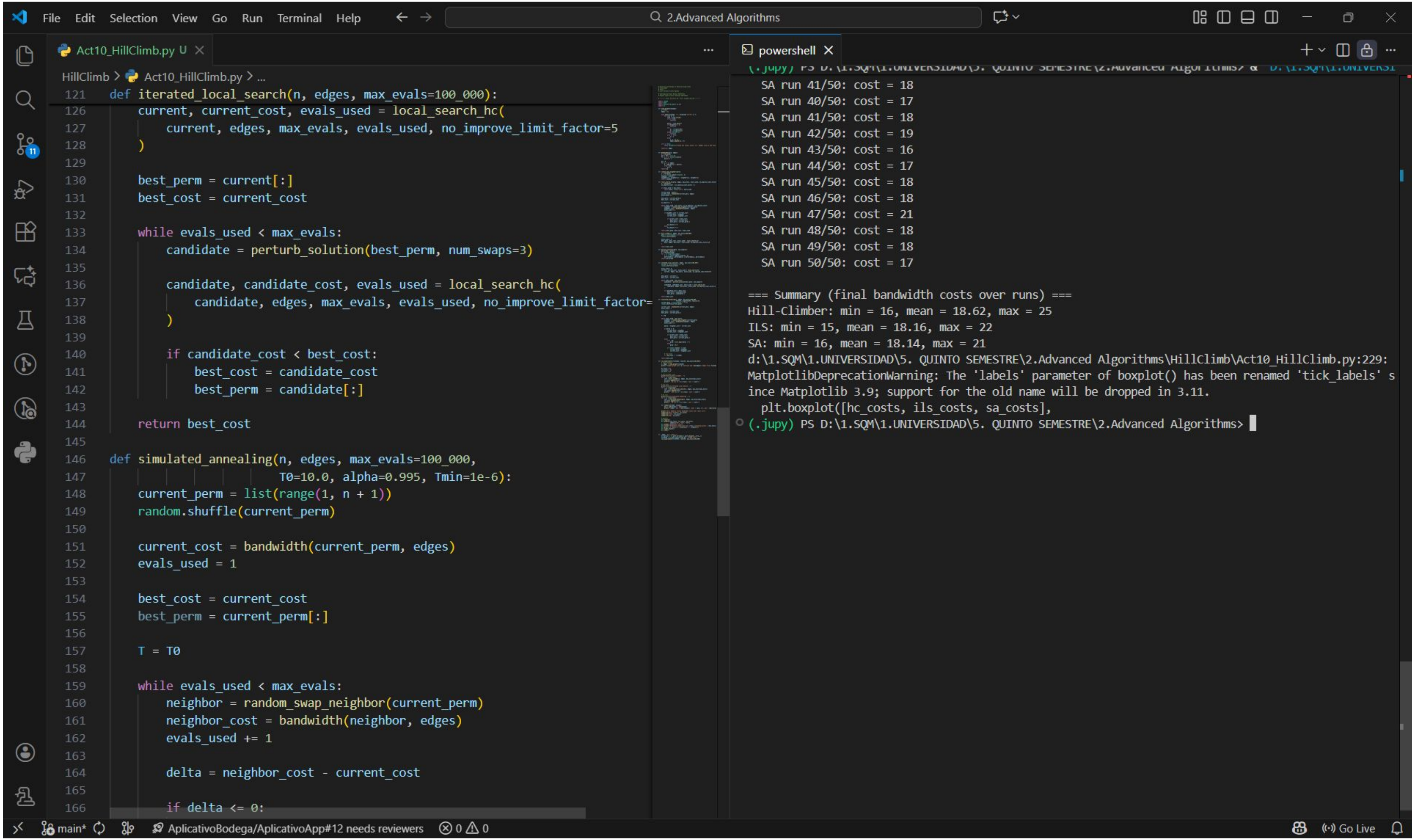
```python
def local_search_hc(perm, edges, max_evals, evals_used, no_improve_limit_fact

        if neighbor_cost <= current_cost:
            current_perm = neighbor
            current_cost = neighbor_cost

            if current_cost < best_cost:
                best_cost = current_cost
                best_perm = current_perm[:]

            no_improve = 0
        else:
            no_improve += 1

    return best_perm, best_cost, evals_used

def hill_climber(n, edges, max_evals=100_000):
    perm = list(range(1, n + 1))
    random.shuffle(perm)

    evals_used = 0
    best_perm, best_cost, evals_used = local_search_hc(
        perm, edges, max_evals, evals_used, no_improve_limit_factor=10
    )

    return best_cost

def perturb_solution(perm, num_swaps=3):
    perturbed = perm[:]
    n = len(perturbed)
    for _ in range(num_swaps):
        i, j = random.sample(range(n), 2)
        perturbed[i], perturbed[j] = perturbed[j], perturbed[i]
    return perturbed


def iterated_local_search(n, edges, max_evals=100_000):
    current = list(range(1, n + 1))
    random.shuffle(current)

    evals_used = 0
    current, current_cost, evals_used = local_search_hc(
```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
  ILS run 50/50: cost = 22
Running Simulated Annealing...
  SA run 1/50: cost = 18
  SA run 2/50: cost = 18
  SA run 3/50: cost = 18
  SA run 4/50: cost = 17
  SA run 5/50: cost = 20
  SA run 6/50: cost = 19
  SA run 7/50: cost = 19
  SA run 8/50: cost = 20
  SA run 9/50: cost = 16
  SA run 10/50: cost = 18
  SA run 11/50: cost = 17
  SA run 12/50: cost = 16
  SA run 13/50: cost = 17
  SA run 14/50: cost = 19
  SA run 15/50: cost = 19
  SA run 16/50: cost = 17
  SA run 17/50: cost = 17
  SA run 18/50: cost = 20
  SA run 19/50: cost = 19
  SA run 20/50: cost = 17
  SA run 21/50: cost = 17
  SA run 22/50: cost = 19
  SA run 23/50: cost = 17
  SA run 24/50: cost = 21
  SA run 25/50: cost = 18
  SA run 26/50: cost = 18
  SA run 27/50: cost = 16
  SA run 28/50: cost = 17
  SA run 29/50: cost = 19
  SA run 30/50: cost = 18
  SA run 31/50: cost = 21
  SA run 32/50: cost = 20
  SA run 33/50: cost = 18
  SA run 34/50: cost = 17
  SA run 35/50: cost = 18
  SA run 36/50: cost = 18
  SA run 37/50: cost = 18
  SA run 38/50: cost = 19
  SA run 39/50: cost = 20
  SA run 40/50: cost = 17
  SA run 41/50: cost = 18
  SA run 40/50: cost = 17
  SA run 41/50: cost = 18
  SA run 42/50: cost = 19
```



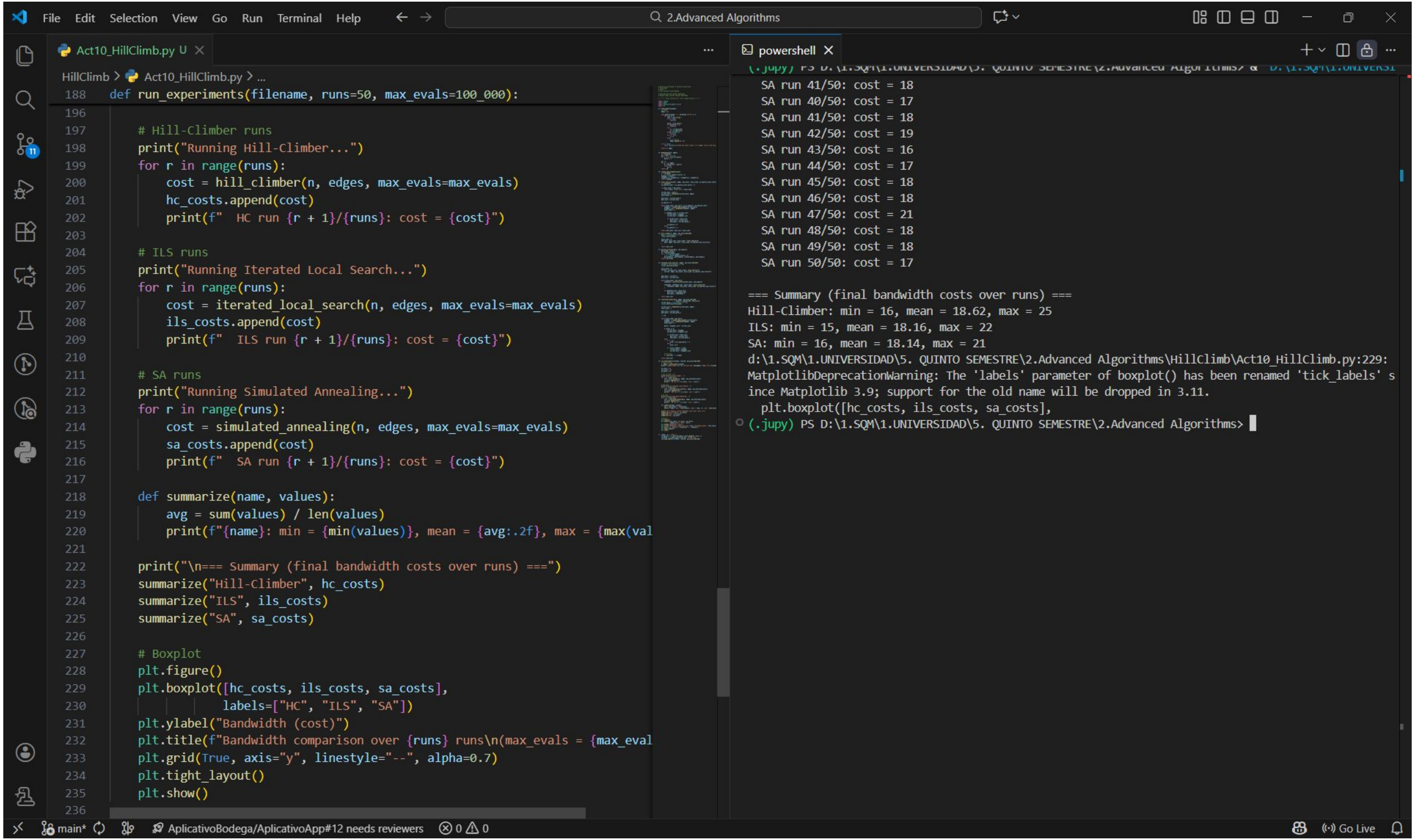
```python
def iterated_local_search(n, edges, max_evals=100_000):
    current, current_cost, evals_used = local_search_hc(
        current, edges, max_evals, evals_used, no_improve_limit_factor=5
    )

    best_perm = current[:]
    best_cost = current_cost

    while evals_used < max_evals:
        candidate = perturb_solution(best_perm, num_swaps=3)

        candidate, candidate_cost, evals_used = local_search_hc(
            candidate, edges, max_evals, evals_used, no_improve_limit_factor=
        )

        if candidate_cost < best_cost:
            best_cost = candidate_cost
            best_perm = candidate[:]

    return best_cost

def simulated_annealing(n, edges, max_evals=100_000,
                        T0=10.0, alpha=0.995, Tmin=1e-6):
    current_perm = list(range(1, n + 1))
    random.shuffle(current_perm)

    current_cost = bandwidth(current_perm, edges)
    evals_used = 1

    best_cost = current_cost
    best_perm = current_perm[:]

    T = T0

    while evals_used < max_evals:
        neighbor = random_swap_neighbor(current_perm)
        neighbor_cost = bandwidth(neighbor, edges)
        evals_used += 1

        delta = neighbor_cost - current_cost

        if delta <= 0:
```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
  SA run 41/50: cost = 18
  SA run 40/50: cost = 17
  SA run 41/50: cost = 18
  SA run 42/50: cost = 19
  SA run 43/50: cost = 16
  SA run 44/50: cost = 17
  SA run 45/50: cost = 18
  SA run 46/50: cost = 18
  SA run 47/50: cost = 21
  SA run 48/50: cost = 18
  SA run 49/50: cost = 18
  SA run 50/50: cost = 17

=== Summary (final bandwidth costs over runs) ===
Hill-Climber: min = 16, mean = 18.62, max = 25
ILS: min = 15, mean = 18.16, max = 22
SA: min = 16, mean = 18.14, max = 21
d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\HillClimb\Act10_HillClimb.py:229: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be dropped in 3.11.
  plt.boxplot([hc_costs, ils_costs, sa_costs],
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
```



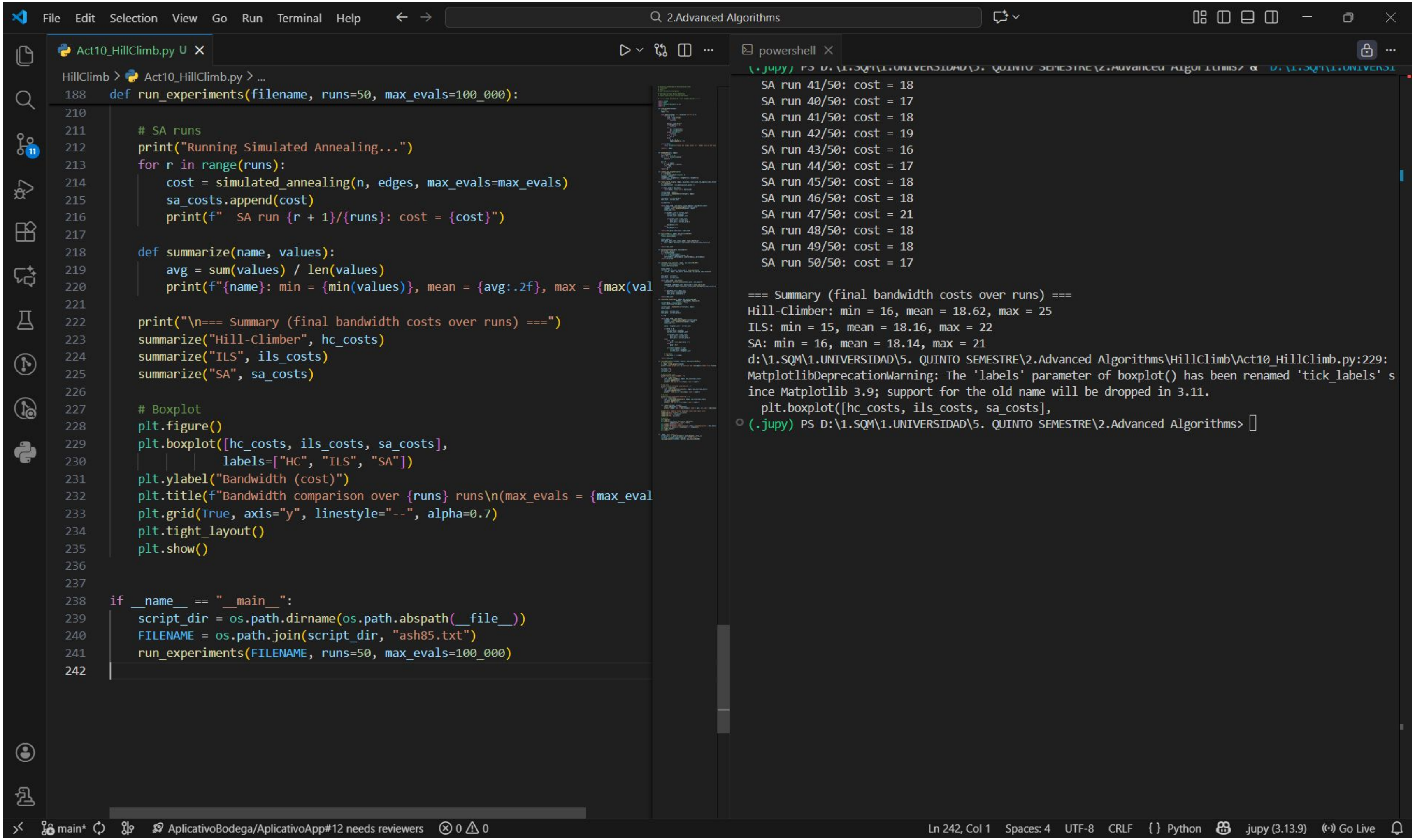
```python
def iterated_local_search(n, edges, max_evals=100_000):
    current, current_cost, evals_used = local_search_hc(
        current, edges, max_evals, evals_used, no_improve_limit_factor=5
    )

    best_perm = current[:]
    best_cost = current_cost

    while evals_used < max_evals:
        candidate = perturb_solution(best_perm, num_swaps=3)

        candidate, candidate_cost, evals_used = local_search_hc(
            candidate, edges, max_evals, evals_used, no_improve_limit_factor=
        )

        if candidate_cost < best_cost:
            best_cost = candidate_cost
            best_perm = candidate[:]

    return best_cost

def simulated_annealing(n, edges, max_evals=100_000,
                        T0=10.0, alpha=0.995, Tmin=1e-6):
    current_perm = list(range(1, n + 1))
    random.shuffle(current_perm)

    current_cost = bandwidth(current_perm, edges)
    evals_used = 1

    best_cost = current_cost
    best_perm = current_perm[:]

    T = T0

    while evals_used < max_evals:
        neighbor = random_swap_neighbor(current_perm)
        neighbor_cost = bandwidth(neighbor, edges)
        evals_used += 1

        delta = neighbor_cost - current_cost

        if delta <= 0:
```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
  SA run 41/50: cost = 18
  SA run 40/50: cost = 17
  SA run 41/50: cost = 18
  SA run 42/50: cost = 19
  SA run 43/50: cost = 16
  SA run 44/50: cost = 17
  SA run 45/50: cost = 18
  SA run 46/50: cost = 18
  SA run 47/50: cost = 21
  SA run 48/50: cost = 18
  SA run 49/50: cost = 18
  SA run 50/50: cost = 17

=== Summary (final bandwidth costs over runs) ===
Hill-Climber: min = 16, mean = 18.62, max = 25
ILS: min = 15, mean = 18.16, max = 22
SA: min = 16, mean = 18.14, max = 21
d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\HillClimb\Act10_HillClimb.py:229:
MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' s
ince Matplotlib 3.9; support for the old name will be dropped in 3.11.
  plt.boxplot([hc_costs, ils_costs, sa_costs],
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
```


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Act10_HillClimb.py U
HillClimb > Act10_HillClimb.py > ...
powershell
main*
AplicativoBodega/AplicativoApp#12 needs reviewers
⊗ 0 ⚠ 0
Go Live

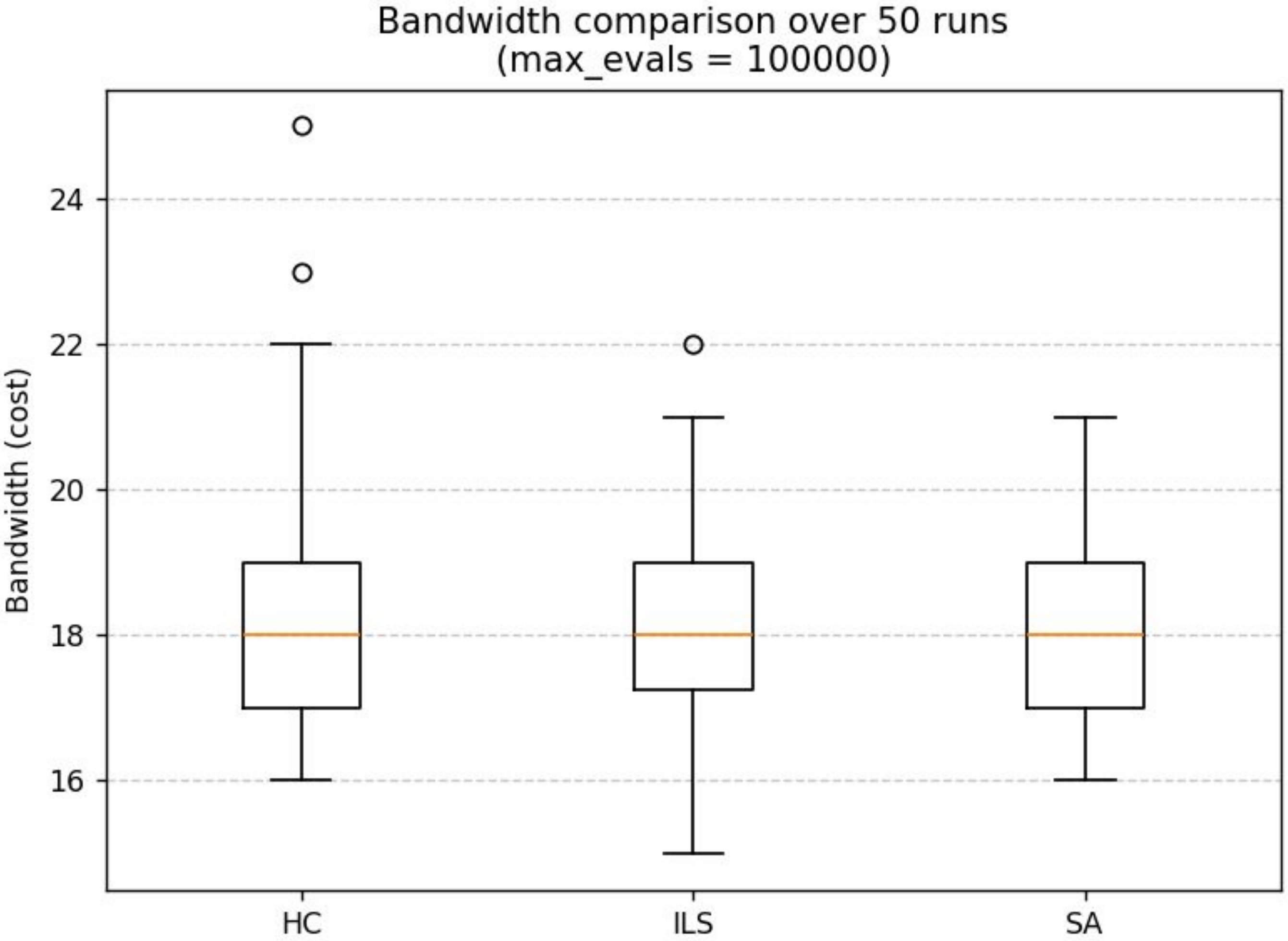
```python
def run_experiments(filename, runs=50, max_evals=100_000):

    # Hill-Climber runs
    print("Running Hill-Climber...")
    for r in range(runs):
        cost = hill_climber(n, edges, max_evals=max_evals)
        hc_costs.append(cost)
        print(f"  HC run {r + 1}/{runs}: cost = {cost}")

    # ILS runs
    print("Running Iterated Local Search...")
    for r in range(runs):
        cost = iterated_local_search(n, edges, max_evals=max_evals)
        ils_costs.append(cost)
        print(f"  ILS run {r + 1}/{runs}: cost = {cost}")

    # SA runs
    print("Running Simulated Annealing...")
    for r in range(runs):
        cost = simulated_annealing(n, edges, max_evals=max_evals)
        sa_costs.append(cost)
        print(f"  SA run {r + 1}/{runs}: cost = {cost}")

    def summarize(name, values):
        avg = sum(values) / len(values)
        print(f"{name}: min = {min(values)}, mean = {avg:.2f}, max = {max(val

    print("\n=== Summary (final bandwidth costs over runs) ===")
    summarize("Hill-Climber", hc_costs)
    summarize("ILS", ils_costs)
    summarize("SA", sa_costs)

    # Boxplot
    plt.figure()
    plt.boxplot([hc_costs, ils_costs, sa_costs],
                labels=["HC", "ILS", "SA"])
    plt.ylabel("Bandwidth (cost)")
    plt.title(f"Bandwidth comparison over {runs} runs\n(max_evals = {max_eval
    plt.grid(True, axis="y", linestyle="--", alpha=0.7)
    plt.tight_layout()
    plt.show()

```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
  SA run 41/50: cost = 18
  SA run 40/50: cost = 17
  SA run 41/50: cost = 18
  SA run 42/50: cost = 19
  SA run 43/50: cost = 16
  SA run 44/50: cost = 17
  SA run 45/50: cost = 18
  SA run 46/50: cost = 18
  SA run 47/50: cost = 21
  SA run 48/50: cost = 18
  SA run 49/50: cost = 18
  SA run 50/50: cost = 17

=== Summary (final bandwidth costs over runs) ===
Hill-Climber: min = 16, mean = 18.62, max = 25
ILS: min = 15, mean = 18.16, max = 22
SA: min = 16, mean = 18.14, max = 21
d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\HillClimb\Act10_HillClimb.py:229:
MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' s
ince Matplotlib 3.9; support for the old name will be dropped in 3.11.
  plt.boxplot([hc_costs, ils_costs, sa_costs],
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
```



```python
def run_experiments(filename, runs=50, max_evals=100_000):

    # SA runs
    print("Running Simulated Annealing...")
    for r in range(runs):
        cost = simulated_annealing(n, edges, max_evals=max_evals)
        sa_costs.append(cost)
        print(f"  SA run {r + 1}/{runs}: cost = {cost}")

    def summarize(name, values):
        avg = sum(values) / len(values)
        print(f"{name}: min = {min(values)}, mean = {avg:.2f}, max = {max(val

    print("\n=== Summary (final bandwidth costs over runs) ===")
    summarize("Hill-Climber", hc_costs)
    summarize("ILS", ils_costs)
    summarize("SA", sa_costs)

    # Boxplot
    plt.figure()
    plt.boxplot([hc_costs, ils_costs, sa_costs],
                labels=["HC", "ILS", "SA"])
    plt.ylabel("Bandwidth (cost)")
    plt.title(f"Bandwidth comparison over {runs} runs\n(max_evals = {max_eval
    plt.grid(True, axis="y", linestyle="--", alpha=0.7)
    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    script_dir = os.path.dirname(os.path.abspath(__file__))
    FILENAME = os.path.join(script_dir, "ash85.txt")
    run_experiments(FILENAME, runs=50, max_evals=100_000)

```

```
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & D:\1.SQM\1.UNIVERSI
  SA run 41/50: cost = 18
  SA run 40/50: cost = 17
  SA run 41/50: cost = 18
  SA run 42/50: cost = 19
  SA run 43/50: cost = 16
  SA run 44/50: cost = 17
  SA run 45/50: cost = 18
  SA run 46/50: cost = 18
  SA run 47/50: cost = 21
  SA run 48/50: cost = 18
  SA run 49/50: cost = 18
  SA run 50/50: cost = 17

=== Summary (final bandwidth costs over runs) ===
Hill-Climber: min = 16, mean = 18.62, max = 25
ILS: min = 15, mean = 18.16, max = 22
SA: min = 16, mean = 18.14, max = 21
d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\HillClimb\Act10_HillClimb.py:229:
MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' s
ince Matplotlib 3.9; support for the old name will be dropped in 3.11.
  plt.boxplot([hc_costs, ils_costs, sa_costs],
(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
```


Bandwidth comparison over 50 runs
(max_evals = 100000)
Bandwidth (cost)
24
22
20
18
16
HC
ILS
SA

https://colab.research.google.com/drive/1PABqw6sBAD4zyjQsSc7sWASRUbfKVyZa?usp=sharing