



FREESTYLE SUBMISSION

Optimizing Party Planning

Mihir Paithane
Mihir.paithane@duke.edu

Explanation of your algorithm:

The goal was to optimize the food and drink choices, and I made the assumption that you will want to provide a food and drink, even if not preferred, to all party attendees.

The Algorithm

1. Find cheapest combo of food and drink overall; call that price GLCC – Global Lowest Costing Combo.
2. Find individual's lowest preferred food and drink combo; call that price ILCPC – Individual Lowest Costing Preferred Combo
3. Ensure the budget is sufficient to provide something i.e. $\text{budget} \geq N \text{ (No. of People)} * \text{cost of GLCC}$, which we will call MinSufficientBudget.
4. Assign best choice for each individual
 - a. Identify what we have left to go over from the absolute minimum for each person i.e. $\text{PreferenceBudget} = \text{Budget} - \text{MinSufficientBudget}$
 - b. Now, sort all individuals such that ILCPC is in ascending order i.e. the lowest preferred items are at the top.
 - c. Walk down that list assigning each person their preference until the PreferenceBudget runs out
 - d. Assign the remaining people the GLCC combo
5. Print the results

For more information regarding the algorithm, refer to “More Details on the Algorithm” at the bottom of this document.

Test cases:

The test cases are split into categories based on the budget amounts.

- Global Min = Minimum budget required to give every party attendee the cheapest food and drink combination
- Preferred Min = Minimum budget required to give every party attendee their cheapest preferred food and drink combination

All the following test cases were run with preferences file “people0.txt”, food file “foods.txt”, and drinks file “drinks.txt”. The food and drinks files can be found in the “Testing” folder of the repository. The preferences file can be found within the “Test Case Preferences” folder inside “Testing”.

1) Budget < Global Min

This test case was important to ensure that the algorithm correctly determined that the budget for the party was not enough to even assign each party attendee the cheapest food and drink combination. Given I made the assumption that every party attendee **must** be provided something, even if not preferred, the algorithm correctly identified that the party cannot be hosted and returned the following screen:

The screenshot shows a web application titled "Party Optimizer Solution - Mihir Paithane". It features a light blue background with a white central panel. At the top of the panel, there is a text input field labeled "Enter the budget:" containing the value "0.0". Below this are four buttons: "Upload Preferences", "Upload Food", "Upload Drinks", and "Optimize". The main content area is titled "PARTY AND GUEST LOGISTICS" and contains the following text: "Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination." followed by a dashed line and "Solution Summary". Below the summary, it states "Budget: \$0.0" and "Given your budget, this party cannot be planned."

2) Budget = Global Min

This test case was important to ensure that the algorithm correctly provided every attendee the overall cheapest food and drink combination because that is the only set of food and drink combinations that the budget allowed. The output for this can be seen here:

The screenshot shows a web application titled "Party Optimizer Solution - Mihir Paithane". It features a light blue background with a central white box containing the results. At the top, there's a section for input: "Enter the budget:" with a text box containing "450.0", and four buttons: "Upload Preferences", "Upload Food", "Upload Drinks", and "Optimize". Below this, the results are displayed under the heading "PARTY AND GUEST LOGISTICS". The first line states: "Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination." This is followed by a dashed line and the "Solution Summary" section, which lists: "Budget: \$450.0", "Total Spent: \$449.1", "Total Money Left: \$0.9", and "Total Preferences Met: 13 out of 100". Another dashed line leads to the "Food Information" section, which lists: "Food Needed:", "92x Breadstick (\$3.4/ea)", "2x Chicken Pesto (\$4.5/ea)", "6x Spaghetti (\$4.5/ea)", and "Total Food Cost: \$348.8". A final dashed line leads to the "Drink Information" section, which lists: "Drinks Needed:", "3x Sweetened Ice Tea (\$1.1/ea)", "97x Orange Juice (\$1.0/ea)", and "Total Drink Cost: \$100.3".

Party Optimizer Solution - Mihir Paithane

Enter the budget:

450.0

Upload Preferences

Upload Food

Upload Drinks

Optimize

PARTY AND GUEST LOGISTICS

Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination.

-----Solution Summary-----

Budget: \$450.0
Total Spent: \$449.1
Total Money Left: \$0.9
Total Preferences Met: 13 out of 100

-----Food Information-----

Food Needed:
92x Breadstick (\$3.4/ea)
2x Chicken Pesto (\$4.5/ea)
6x Spaghetti (\$4.5/ea)
Total Food Cost: \$348.8

-----Drink Information-----

Drinks Needed:
3x Sweetened Ice Tea (\$1.1/ea)
97x Orange Juice (\$1.0/ea)
Total Drink Cost: \$100.3

3) (Budget > Global Min) and (Budget < Preferred Min)

This test case was particularly important to ensure that the algorithm assigned certain attendees their preferred lowest costing combination and certain attendees the overall lowest costing combination. This can be seen to be true based on the fact that a portion of the total attendee's preferences were met, but not all were met. Also, notice for increase budget values the number of preferences met increases, signifying that the algorithm is correctly able to afford an increasing number of party attendees' lowest costing preferred food and drink combinations. Output:

Party Optimizer Solution - Mihir Palthane

Enter the budget:
645

Upload Preferences
Upload Food
Upload Drinks
Optimize

PARTY AND GUEST LOGISTICS

Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination.

-----Solution Summary-----
Budget: \$645.0
Total Spent: \$640.4
Total Money Left: \$4.6
Total Preferences Met: 72 out of 100

-----Food Information-----
Food Needed:
3x Grilled Cheese (\$9.7/ea)
53x Breadstick (\$3.4/ea)
3x BLT (\$8.7/ea)
19x Chicken Pesto (\$4.5/ea)
16x Spaghetti (\$4.5/ea)
6x Pizza (\$6.9/ea)
Total Food Cost: \$434.3

-----Drink Information-----
Drinks Needed:
12x Sprite (\$2.3/ea)

Party Optimizer Solution - Mihir Palthane

Enter the budget:
800

Upload Preferences
Upload Food
Upload Drinks
Optimize

PARTY AND GUEST LOGISTICS

Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination.

-----Solution Summary-----
Budget: \$800.0
Total Spent: \$796.3
Total Money Left: \$3.7
Total Preferences Met: 92 out of 100

-----Food Information-----
Food Needed:
5x Grilled Cheese (\$9.7/ea)
1x Chicken Parmesan (\$10.3/ea)
33x Breadstick (\$3.4/ea)
6x BLT (\$8.7/ea)
19x Chicken Pesto (\$4.5/ea)
3x Vegetable Lasagna (\$10.0/ea)
11x Fettucini Alfredo (\$10.0/ea)
16x Spaghetti (\$4.5/ea)
6x Pizza (\$6.9/ea)
Total Food Cost: \$562.1

4) Budget = Preferred Min

This test was important to ensure that the algorithm correctly determined that there was enough money to assign all attendees one of their preferred food and drink combinations. Given that all the attendees' preferences were met, the algorithm correctly concluded that all attendees can be given their preferred combinations.

Output:

The screenshot shows a web application titled "Party Optimizer Solution - Mihir Paithane". It features a light blue background with a central white box containing the results. At the top, there is a form with a text input field labeled "Enter the budget:" containing the value "883.8". Below this are four buttons: "Upload Preferences", "Upload Food", "Upload Drinks", and "Optimize". The results section is titled "PARTY AND GUEST LOGISTICS" and contains the following text:

Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination.

-----Solution Summary-----

Budget: \$883.8
Total Spent: \$883.8
Total Money Left: \$0
Total Preferences Met: 100 out of 100

-----Food Information-----

Food Needed:
7x Grilled Cheese (\$9.7/ea)
3x Chicken Parmesan (\$10.3/ea)
25x Breadstick (\$3.4/ea)
6x BLT (\$8.7/ea)
19x Chicken Pesto (\$4.5/ea)
1x Black Bean Burger (\$17.8/ea)
1x Thai Curry (\$14.0/ea)
3x Vegetable Lasagna (\$10.0/ea)
13x Fettucini Alfredo (\$10.0/ea)
16x Spaghetti (\$4.5/ea)
6x Pizza (\$6.9/ea)

5) Budget > Preferred Min

This test was important to ensure that the algorithm correctly assigned all attendees one of their preferred food and drink combinations with money left over. Given that the “Total Money Left” was greater than 0, the algorithm was correct for this test case. Also, notice that for greater budget values, the money left over increase, which signifies the algorithm accurately does not spend that money on more expensive items. Output:

Party Optimizer Solution - Mihir Paithane

Enter the budget:
1000.0

Upload Preferences
Upload Food
Upload Drinks
Optimize

PARTY AND GUEST LOGISTICS

Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination.

-----Solution Summary-----

Budget: \$1000.0
Total Spent: \$883.8
Total Money Left: \$116.2
Total Preferences Met: 100 out of 100

-----Food Information-----

Food Needed:
7x Grilled Cheese (\$9.7/ea)
3x Chicken Parmesan (\$10.3/ea)
25x Breadstick (\$3.4/ea)
6x BLT (\$8.7/ea)
19x Chicken Pesto (\$4.5/ea)
1x Black Bean Burger (\$17.8/ea)
1x Thai Curry (\$14.0/ea)
3x Vegetable Lasagna (\$10.0/ea)
13x Fettucini Alfredo (\$10.0/ea)
16x Spaghetti (\$4.5/ea)
6x Pizza (\$6.9/ea)

Party Optimizer Solution - Mihir Paithane

Enter the budget:
5000.0

Upload Preferences
Upload Food
Upload Drinks
Optimize

PARTY AND GUEST LOGISTICS

Minimum budget of \$883.8 is required to accommodate all attendees with their cheapest preferred food/drink combination.

-----Solution Summary-----

Budget: \$5000.0
Total Spent: \$883.8
Total Money Left: \$4116.2
Total Preferences Met: 100 out of 100

-----Food Information-----

Food Needed:
7x Grilled Cheese (\$9.7/ea)
3x Chicken Parmesan (\$10.3/ea)
25x Breadstick (\$3.4/ea)
6x BLT (\$8.7/ea)
19x Chicken Pesto (\$4.5/ea)
1x Black Bean Burger (\$17.8/ea)
1x Thai Curry (\$14.0/ea)
3x Vegetable Lasagna (\$10.0/ea)
13x Fettucini Alfredo (\$10.0/ea)
16x Spaghetti (\$4.5/ea)
6x Pizza (\$6.9/ea)

Assumptions:

1. The party planner will want to provide at least something, even if it is not preferred, to all party attendees. Therefore, a food and drink combination should be provided for every party attendee, otherwise the party cannot be planned.
2. Each party attendee only needs one drink item.
3. Each party attendee only needs one food item.

More Details on the Algorithm:

This is a description of the algorithm with more details. Following the goal of optimizing the food and drink choices, and the assumptions that you will want to provide a food and drink, even if not preferred, to all party attendees, my algorithm will first determine the overall cheapest (lowest costing) food and drink combo that exists within all the possible food and drink items that can be bought for the party. Picking the cheapest food and drink item represents the lowest costing food and drink combo (referred to as GLCC for Global Lowest Costing Combo). The combination of these two represents the cheapest possible food and drink combination that can be given to all attendees.

Then, the algorithm proceeds to determine the cheapest food and drink combination for each party attendee based on their preferences (referred to as ILCPC for Individual Lowest Costing Preferred Combo). It establishes the ILCPC for each attendee by choosing the cheapest (per unit cost) food and drink item in the attendee's preferences, thus creating the ILCPC for the individual. This list of ILCPC's for all party attendees is used later in the algorithm.

However, the algorithm checks to see, given the budget, if all the party attendees can be given the GLCC. It checks this by determining if the budget is less than the cost of the GLCC * the number of party attendees. If it is, the algorithm states that this party cannot be held since there is not enough money to give each attendee at least the GLCC. If the budget is enough to at least assign the GLCC to every party attendee, the algorithm continues.

Assuming the aforementioned budget check was passed, the algorithm determines how much money is remaining if the GLCC was simply assigned to all attendees. This remaining budget is calculated by subtracting the cost of the GLCC * the number of party attendees from the overall given budget.

This remaining budget is signified as the "rem_budget" in the algorithm. Using this leftover money, the algorithm systematically chooses, starting from the cheapest ILCPC to the most expensive ILCPC, if there are ILCPCs that can be assigned to individuals instead of simply giving them the GLCC, which they may not prefer. The algorithm determines this by adding the already allotted cost of the GLCC and the leftover money, and then seeing if an ILCPC is less than this amount (implying that the algorithm can afford to give an individual one of their preferred food and drink combinations over simply giving them the GLCC). If the algorithm can afford this ILCPC, it assigns the individual their preferred combination and decreases the remaining budget by the difference in cost of the ILCPC and the GLCC (the cost of choosing the preferred combo over the base combo). However, in the case where the algorithm cannot afford to give the ILCPC of an individual, it simply assigns the GLCC to the individual. It does not decrement the remaining budget since it could not afford to decrease the remaining budget.

These set of steps are performed for all the ILPCs, and thus results in assigning all attendees with either their lowest costing preferred food and drink combination or the overall lowest costing food and drink combination.