

Algorithm Design-II(CSE 4131)  
TERM PROJECT REPORT  
(March'2023-July'2023)  
ON

***Social Event Scheduling Using  
Greedy Approach***

Submitted by

***Mansika Sahoo***

Registration No: 2141019112

B.Tech. 4<sup>th</sup> Semester CSE 'U'



*Department of Computer Science and Engineering  
Institute of Technical Education and Research  
Siksha 'O' Anusandhan Deemed To Be University  
Bhubaneswar, Odisha-751030*

# DECLARATION

I, **Mansika Sahoo**, bearing the registration number 2141019112 do hereby declare that this “**Social Event Scheduling Using Greedy Approach**” term project entitled is an original project work done by me and has not been previously submitted to any university or research institution or department for the award of any degree or diploma or any other assessment to the best of my knowledge.

**Name: Mansika Sahoo**

**Regd. No.: 2141019112**

**Date: 14.06.23**

# CERTIFICATE

*This is to certify that the thesis entitled “**Social Event Scheduling Using Greedy Approach**” submitted by **Mansika Sahoo**, bearing registration number 2141019112 of B.Tech. 4th Semester Comp. Sc. and Engg., ITER, SOADU is absolutely based upon his/her own work under my guidance and supervision.*

*The term project has reached the standard fulfilling the requirement of the course Algorithm Design 2 (CSE4131). Any help or source of information which has been available in this connection is duly acknowledged.*

***Subha Mondal***

*Assistant Professor,  
Department of Comp. Sc. and Engg.  
ITER, Bhubaneswar 751030,  
Odisha, India*

*Prof. (Dr.) Debahuti Mishra  
Professor and Head,  
Department of Comp. Sc. and Engg.  
ITER, Bhubaneswar 751030,  
Odisha, India*

# ABSTRACT

*Effective activity scheduling in social events, such as weddings and parties, is vital for guest enjoyment and host stress reduction. This abstract highlights the significance of thoughtful planning to create memorable experiences while alleviating the host's burden. Well-structured event schedules balance activities, transitions, and breaks, ensuring a smooth flow and preventing disengagement or discomfort. Maximizing guest enjoyment involves offering diverse and interactive elements, considering energy levels and pacing for a vibrant experience.*

*Advancements in event management technology and communication tools have facilitated more efficient activity scheduling. Digital platforms offer features such as customizable timelines, real-time updates, and automated reminders that aid in streamlining the planning process. These tools enable hosts to collaborate with vendors, share event details with guests, and make adjustments to the schedule as needed, enhancing overall organization and coordination. In conclusion, the strategic scheduling of activities in social events plays a crucial role in maximizing guest enjoyment and reducing host stress. By thoughtfully planning the event timeline, hosts can create a seamless and engaging experience that caters to the preferences and energy levels of attendees. Simultaneously, efficient scheduling allows hosts to manage their responsibilities more effectively and enjoy the event alongside their guests.*

# **CONTENTS**

## **1. INTRODUCTION**

- **OVERVIEW**
- **PROBLEM DESCRIPTION**
- **PROBLEM STATEMENT**
- **MATHEMATIC FORMULATION**
- **ASSUMPTION**

## **2. DESIGNING ALGORITHM**

- **PSEUDOCODE**
- **DESCRIPTION OF STEPS**
- **EXAMPLES**

## **3. IMPLEMENTATION DETAILS**

## **4. RESULTS AND DISCUSSION**

## **5. LIMITATIONS**

## **6. FUTURE ENHANCEMENTS**

## **7. REFERENCES**

# 1.INTRODUCTION

## Overview:

*In social events like weddings or parties, scheduling activities is crucial to ensure maximum guest enjoyment and minimize the stress experienced by the host.*

### 1. Importance of activity scheduling:

- Activity scheduling plays a vital role in creating a memorable and enjoyable experience for guests.*
- It helps in organizing the event and maintaining a smooth flow of activities.*

### 2. Factors to consider in event scheduling:

- Purpose and duration of the event: Understanding the objective and timeframe helps in planning appropriate activities.*
- Guest demographics: Consideration of guests' preferences, age groups, and cultural backgrounds ensures inclusivity.*

### 3. Balancing activities, transitions, and breaks:

- A well-structured schedule maintains a balance between different activities, ensuring guests are engaged and entertained.*
- Thoughtful transitions between activities prevent lulls or overcrowded periods, enhancing the overall experience.*

### 4. Maximizing guest enjoyment:

- Offering a diverse range of interactive elements such as games, performances, speeches, and dancing caters to different interests.
- Sequencing activities based on energy levels and pacing helps maintain a dynamic and engaging atmosphere.

#### **5. Minimizing host stress:**

- Effective activity scheduling allows hosts to allocate their time and resources efficiently, reducing last-minute disruptions.
- Building buffer time in the schedule provides flexibility to handle unexpected situations and relieves the host's stress.

### **Problem Description:**

**Real life Scenario:** Jane is planning her wedding reception and wants to incorporate engaging activities to ensure her guests have a memorable experience. She is looking for a variety of options that cater to different age groups and interests. Here's how the Activity Library App can assist Jane:

#### **1. Extensive Activity Collection:**

- Jane opens the app and discovers a wide range of activities specifically curated for wedding receptions.
- The library includes options like photo booths, lawn games, interactive trivia, dance tutorials, and live musical performances.

#### **2. Activity Descriptions and Instructions:**

- Jane selects a specific activity, such as the photo booth, and reads its description and instructions provided by the app.

- She learns about the necessary equipment, setup requirements, and how guests can engage with the activity.

### **3. Recommendations and Filters:**

- The app recommends activities based on the event type and guest demographics, taking into account factors like age groups, interests, and indoor/outdoor preferences.

- Jane applies filters to narrow down her choices to activities suitable for both adults and children, ensuring everyone can participate and enjoy.

### **4. User Reviews and Ratings:**

- Jane checks the user reviews and ratings for the activities she's interested in.

- She finds positive feedback from other users who have implemented the activities at their own weddings, giving her confidence in their success.

### **5. Activity Customization:**

- Jane decides to customize the trivia game activity to incorporate fun facts about her and her partner.

- She modifies the questions to be more personalized, making the activity unique to her wedding reception.

By using the Activity Library App, Jane can efficiently browse through a wide range of activities specifically curated for weddings. She can select activities based on recommendations, read detailed descriptions and instructions, customize them to align with her event, and gain insights from user reviews. This app



*streamlines the process of finding and implementing engaging activities, ensuring Jane's guests have an enjoyable and memorable experience at her wedding reception.*

### **Problem Statement:**

*In social events such as weddings or parties, hosts often struggle to find suitable and engaging activities, games, performances, or entertainment options that align with their event vision. The lack of a centralized resource for diverse and pre-defined activities poses a significant challenge. Without access to an extensive activity library, hosts may face difficulties in creating an entertaining and memorable experience for their guests. They often spend considerable time and effort searching for activities that cater to different age groups, interests, and event themes.*

*The consequences of inadequate activity selection can be disengaged guests, a lackluster event atmosphere, and a diminished overall experience. In addition, hosts may feel overwhelmed and stressed when faced with the challenge of brainstorming and organizing suitable activities that seamlessly fit into the event timeline.*

*Therefore, there is a pressing need for a solution that provides a comprehensive activity library. Such a resource would empower hosts to browse through a wide range of pre-defined activities, games, performances, and entertainment options suitable for different event types. By having access to a diverse array of choices, hosts can easily select activities that align with their event vision, cater to guest preferences,*

*and ensure an enjoyable and unforgettable experience for all attendees.*

## **Mathematical Formulation:**

*Let's represent the problem as a mathematical optimization model using a greedy algorithm approach. We aim to maximize the overall enjoyment of guests while minimizing the stress of the host by scheduling activities in social events such as weddings or parties.*

*Variables:*

- Let  $A$  be the set of available activities in the activity library.*
- Let  $G$  be the set of guests attending the event.*
- Let  $T$  be the total time available for the event.*

*Objective:*

*Maximize the overall enjoyment of guests.*

*Maximize:  $\sum_{g \in G} \sum_{a \in A} X_{ga} * E_g(a)$*

*Subject to:*

- Each guest should participate in at least one activity.*

$$\forall g \in G: \sum_{a \in A} X_{ga} \geq 1$$

- The total time taken by the scheduled activities should not exceed the available time.*

$$\sum_{a \in A} \sum_{g \in G} X_{ga} * T_a \leq T$$

- Binary decision variables:*

$$X_{ga} = 1 \text{ if activity } a \text{ is scheduled for guest } g$$

$X_{ga} = 0$  otherwise

-  $E_g(a)$  represents the enjoyment factor of activity  $a$ .

**Greedy Algorithm:**

1. Initialize the schedule  $S$  as an empty set.
2. While there are guests without activities:
  - a. For each guest  $g$ , calculate the enjoyment factor  $E_g(a)$  for each available activity  $a$ .
  - b. Select the activity  $a$  with the highest enjoyment factor  $E_g(a)$  for the guest  $g$ .
  - c. Add the selected activity  $a$  to the schedule  $S$ .
  - d. Mark guest  $g$  as having participated in an activity.
3. If the total time of the scheduled activities in  $S$  exceeds  $T$ , remove the activity with the lowest enjoyment factor until the schedule fits within the available time.

The greedy algorithm iteratively selects the highest-enjoyment activities for each guest, ensuring that all guests are assigned activities and maximizing their overall enjoyment. The algorithm then adjusts the schedule to fit within the available time constraints, prioritizing activities with higher enjoyment factors.

## Assumptions:

1. **Activities are independent:** The enjoyment factor of an activity does not depend on the presence or absence of other activities in the schedule. Each activity's

*enjoyment factor is determined solely by its inherent characteristics.*

*2. Fixed duration: The duration of each activity,  $T_a$ , is predetermined and remains constant. It does not vary based on guest participation or other factors.*

*3. Non-overlapping activities: It is assumed that activities do not overlap in time. Guests can participate in one activity at a time, and there are no conflicts or scheduling constraints that prevent them from engaging in multiple activities.*

*4. Unlimited availability: The activity library, contains a wide range of available activities to choose from, and there are no limitations or restrictions on the number of times an activity can be scheduled or the number of guests who can participate in an activity.*

*5. Linear enjoyment relationship: The enjoyment factor,  $E_g(a)$ , is represented as a linear relationship with the activity. It assumes that the enjoyment experienced by a guest is directly proportional to the enjoyment factor associated with the selected activity.*

*6. Single event type: The mathematical formulation assumes a single event type (e.g., weddings or parties). Different event types may have different requirements and preferences, which would require customization of the algorithm and formulation.*

## 2. DESIGNING ALGORITHM

### Pseudocode:

*Input:*

- *A: Set of available activities*
- *G: Set of guests attending the event*
- *T: Total time available for the event*

*Output:*

- *Schedule: Set of scheduled activities*

*Greedy Scheduling(A, G, T):*

*Initialize an empty schedule S*

*while G is not empty:*

*for each guest g in G:*

*Calculate the enjoyment factor  $Eg(a)$  for each available activity a in A*

*Select the activity a with the highest enjoyment factor  $Eg(a)$  for guest g*

*Add activity a to schedule S*

*Mark guest g as having participated in an activity*

*if total time of scheduled activities in S exceeds T:*

*Adjust schedule S to fit within the available time*

*return Schedule S*

## **Description Of Each Steps:**

**1. Initialize an empty schedule  $S$ : Creates an empty set to store the scheduled activities.**

**2. Iterate while there are guests remaining:**

- For each guest  $g$  in  $G$ :**
  - Calculate the enjoyment factor  $Eg(a)$  for each available activity  $a$  in  $A$ : Determines the enjoyment factor of each activity for the current guest.**
  - Select the activity  $a$  with the highest enjoyment factor  $Eg(a)$  for guest  $g$ : Picks the activity with the highest enjoyment factor for the current guest.**
  - Add activity  $a$  to schedule  $S$ : Includes the selected activity in the schedule.**
  - Mark guest  $g$  as having participated in an activity: Updates the status of the guest as having participated in an activity.**

**3. Adjust the schedule if the total time exceeds  $T$ :**

- Checks if the total time of the scheduled activities in  $S$  exceeds the available time  $T$ .**
- If it does, adjusts the schedule by removing activities with the lowest enjoyment factors until the schedule fits within the available time.**

**3. Return the final schedule  $S$ : Provides the set of scheduled activities as the output of the algorithm.**

**Note: The algorithm assumes that the enjoyment factors and schedule adjustment details are implemented elsewhere and are not explicitly described in the pseudocode.**

## **EXAMPLES:**

### **Example 1:**

#### **Input:**

- A: {Dancing, Trivia, Karaoke}
- G: {Guest1 (age: 25), Guest2 (age: 30), Guest3 (age: 28)}
- T: 4 hours

#### **Output:**

- Schedule: {Dancing (Guest1), Trivia (Guest2), Karaoke (Guest3)}

#### **Description:**

- Guest1 (age 25) participates in the Dancing activity, which is suitable for young adults and provides an enjoyable experience.
- Guest2 (age 30) participates in the Trivia activity, which caters to a slightly older age group and offers engaging mental stimulation.
- Guest3 (age 28) participates in the Karaoke activity, allowing them to showcase their singing talents and enjoy a lively interactive experience.
- The total time taken by the scheduled activities is within the available 4 hours, so no adjustment is needed.

### **Example 2:**

#### **Input:**

- A: {Crafts, Puzzle, Magic Show}
- G: {Guest1 (age: 40), Guest2 (age: 35), Guest3 (age: 42)}
- T: 2 hours

#### **Output:**

- *Schedule: {Crafts (Guest1)}*

### *Description:*

- *Guest1 (age 40) participates in the Crafts activity, which is suitable for creative expression and provides a relaxing and enjoyable experience for adults.*
- *Since the total time taken by the scheduled activity (Crafts) exceeds the available 2 hours, the algorithm adjusts the schedule by removing activities with lower enjoyment factors.*
- *In this case, only the Crafts activity fits within the available time, so it remains as the only scheduled activity.*

## **3. IMPLEMENTATION DETAILS**

```
C: > Users > mansi > Downloads > ADCODESES.py
1  class Activity:
2      def __init__(self, name, enjoyment_factor):
3          self.name = name
4          self.enjoyment_factor = enjoyment_factor
5
6
7  class Guest:
8      def __init__(self, name, age):
9          self.name = name
10         self.age = age
11
12
13 def greedy_scheduling(activities, guests, total_time):
14     schedule = []
15
16     # Assign activities to guests based on enjoyment factors
17     while guests:
18         current_guest = guests[0]
19         selected_activity = None
20         max_enjoyment = float('-inf')
21
22         # Find the activity with the maximum enjoyment factor
23         for activity in activities:
24             if activity.enjoyment_factor > max_enjoyment:
25                 max_enjoyment = activity.enjoyment_factor
26                 selected_activity = activity
27
28         # Assign the selected activity to the current guest and remove the guest from the list
29         schedule.append(selected_activity)
30         guests.remove(current_guest)
31
```



C: > Users > mansi > Downloads > ADCODESES.py

```
32     # Adjust the schedule if the total time exceeds the available time
33     if calculate_total_time(schedule) > total_time:
34         adjust_schedule(schedule, total_time)
35
36     return schedule
37 def calculate_total_time(schedule):
38     total_time = 0
39     for activity in schedule:
40         total_time += activity.enjoyment_factor
41     return total_time
42 def adjust_schedule(schedule, total_time):
43     while calculate_total_time(schedule) > total_time:
44         lowest_enjoyment_activity = schedule[0]
45         min_enjoyment = float('inf')
46
47         # Find the activity with the minimum enjoyment factor
48         for activity in schedule:
49             if activity.enjoyment_factor < min_enjoyment:
50                 min_enjoyment = activity.enjoyment_factor
51                 lowest_enjoyment_activity = activity
52
53         # Remove the activity with the lowest enjoyment factor
54         schedule.remove(lowest_enjoyment_activity)
55 def print_schedule(schedule):
56     for activity in schedule:
57         print(activity.name)
58
59 # Example 1
60 activities1 = [
61     Activity("Dancing", 8),
```

FIRST.java    adcode.java    ADCODESES.py

C: > Users > mansi > Downloads > ADCODESES.py

```
61     Activity("Trivia", 9),
62     Activity("Karaoke", 7)
63 ]
64 guests1 = [
65     Guest("Guest1", 25),
66     Guest("Guest2", 30),
67     Guest("Guest3", 28)
68 ]
69 total_time1 = 4
70 # Example 2
71 activities2 = [
72     Activity("Crafts", 7),
73     Activity("Puzzle", 6),
74     Activity("Magic Show", 9)
75 ]
76 guests2 = [
77     Guest("Guest1", 40),
78     Guest("Guest2", 35),
79     Guest("Guest3", 42)
80 ]
81 total_time2 = 2
82 # Schedule activities for Example 1
83 schedule1 = greedy_scheduling(activities1, guests1, total_time1)
84 print("Example 1 Schedule:")
85 print_schedule(schedule1)
86 print()
87 # Schedule activities for Example 2
88 schedule2 = greedy_scheduling(activities2, guests2, total_time2)
89 print("Example 2 Schedule:")
90 print_schedule(schedule2)
```

## ***Data Structures Used:***

*The Python code uses the following data structures:*

- Class: The code defines two classes, 'Activity' and 'Guest', to represent the activities and guests, respectively. Each class has two attributes, 'name' and 'enjoyment\_factor' for 'Activity', and 'name' and 'age' for 'Guest'.*
- List: The code uses lists to represent the activities, guests, and the schedule. The 'activities' and 'guests' lists are initialized with instances of the 'Activity' and 'Guest' classes, respectively. The 'schedule' list is used to store the selected activities.*
- Integer: The code uses integers to represent the total time available for the event, and the age of each guest.*
- Float: The code uses float values to represent the enjoyment factor of each activity.*
- String: The code uses strings to represent the names of the activities and guests.*

*The 'greedy\_scheduling' function uses a greedy algorithm to assign activities to guests based on their enjoyment factors. The 'calculate\_total\_time' function calculates the total time required for the selected activities, and the 'adjust\_schedule' function removes the activity with the lowest enjoyment factor if the total time exceeds the available time.*

## **4. Result and Discussion:**

*The Activity Library app provides hosts with an extensive collection of pre-defined activities, games, performances, and entertainment options for various types of events. Hosts can browse through the library and select activities that align with their event vision. This section discusses the results and benefits of using the Activity Library app.*

*1. Diverse Range of Options: The Activity Library offers a wide range of options, including activities, games, performances, and entertainment options. This ensures that hosts have a diverse selection to choose from, catering to different event types and preferences.*

*2. Time-Saving: By providing a pre-defined library, the app saves hosts valuable time and effort in researching and brainstorming activity ideas. They can quickly explore the options and select activities that suit their event, eliminating the need for extensive planning and ideation.*

*3. Streamlined Event Planning: The Activity Library app streamlines the event planning process by offering a centralized platform to access a wide variety of activities. Hosts no longer need to search for activity ideas from various sources or spend time creating custom activities. They can efficiently select from the pre-defined options provided by the app.*

***4. Customization and Personalization: While the app offers a pre-defined library, hosts still have the flexibility to personalize their events. They can choose activities that best fit their event vision, audience demographics, and theme. This allows for a customized and tailored experience for the guests.***

***5. Enhancing Guest Experience: By selecting activities that align with their event vision, hosts can create an engaging and enjoyable experience for their guests. The Activity Library app empowers hosts to curate an event program that maximizes guest enjoyment, leading to a memorable and successful social gathering.***

***Overall, the Activity Library app simplifies the event planning process by providing a diverse collection of pre-defined activities, games, performances, and entertainment options. It allows hosts to easily browse and select activities that align with their event vision, enhancing the overall guest experience and reducing the stress associated with event planning.***

## 5. LIMITATIONS

*Here are few limitations of the algorithm we used:-*

*1. Limited Pre-defined Activities: The Activity Library app is limited to the activities, games, performances, and entertainment options that are pre-defined within the library. If hosts have specific or unique activity requirements, they may not find suitable options in the library.*

*2. Lack of User-Generated Content: The app currently does not allow users to add their own activities to the library. This limits the customization and personalization options for hosts who have specific activity ideas or want to contribute to the library with their own creations.*

*3. Lack of Activity Ratings and Reviews: The app does not provide ratings or reviews for the pre-defined activities. Hosts might find it helpful to have feedback from other users who have previously used the activities to assess their popularity and suitability.*

*4. Language and Cultural Limitations: The pre-defined activities in the library may be biased towards specific languages, cultures, or regions. This limitation could potentially exclude certain hosts or make it challenging for them to find activities that resonate with their cultural or linguistic context.*

## **6. FUTURE ENHANCEMENTS:**

**1. User-Generated Content:** To enhance the app's versatility, allowing users to contribute their own activities to the library would expand the range of options available. This could be done by implementing a submission and review process for user-generated content.

**2. Activity Recommendations:** The app could incorporate a recommendation system that suggests activities based on the host's event preferences, theme, or guest demographics. This would assist hosts in finding activities that align more closely with their specific needs.

**3. Integration with Event Planning Tools:** The app could be integrated with popular event planning tools, allowing hosts to seamlessly import selected activities into their overall event itinerary. This integration would provide a more comprehensive and streamlined event planning experience.

**5. Social Sharing and Collaboration:** Enabling hosts to share their selected activities and collaborate with other users would foster a sense of community and encourage idea exchange.

## **7. REFERENCES:**

- 1. Kleinberg, J., & Tardos, E. (2006). *Algorithm design*. Pearson Education India.**
- 2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.**
- 3. [www.google.com](http://www.google.com)**
- 4. [ieeexplore.ieee.org](http://ieeexplore.ieee.org)**