

A Mini Project Report
on
FITNESS FREAK
Course: Design and Analysis of Algorithms
Sem: IV Sec: IT-A

By
T.Devi Srujana
1602-20-737-011

T.Vishnu Vardhan
1602-20-737-060



Department of Information technology
Vasavi College of Engineering (Autonomous)
Ibrahimbagh, Hyderabad-31

2022

1. Acknowledgment:

We respect and thank Haseeba ma'am for constant support and motivation which helped us immensely in completing our project successfully.

TABLE OF CONTENTS:

S.No	Content	Page.No
1	Abstract	3
2	Introduction	4-5
3	Design-Strategy,Algorithm, Modules	5-6
4	Code/Implementation	7-12
5	Outputs	12-13
6	Test cases	14-15
7	Conclusion	16
8	References	17

2.PROJECT ABSTRACT:

OBJECTIVE:

The main objective of the project is to maximize the number of calories burnt in a given set of days.

DAYS	1	2	3	4	n
LOW CALORIE	L1	L2	L3	L4	Ln
HIGH CALORIE	H1	H2	H3	H4	Hn

You are allowed to pick either a low calorie exercise or a high calorie exercise; however, picking a high calorie exercise at day i means that you must take the day before (i.e. day $i - 1$) off. Your goal is to maximize the total income. Let's suppose that you are allowed to take the high-calorie exercise in day 1.

Existing system:

In gyms usually people are not aware of how many calories are burnt for a particular exercise. By giving them this information transparency is maintained. One can then decide on his choices as to choose either to go for low or high strain exercises in order to maximize the calories burnt..

Working :

The code will accept the number of days and asks the user to select desired exercises from two categories -high calorie and low calorie.

The code outputs the order of exercises such that the total calories burnt at the end is maximised including the constraint that a person must take rest the before day if he wants to try out the high intense workout the next day.

Concepts used:dynamic programming (maximisation problem), recurrence function.

Data structure:arrays

Language used : C language.

3.Introduction

Nowadays amidst busy lifestyles and resulting problems like obesity many people are now turning to gyms for fitness. Exercises are proven to give best results when different kinds are mixed together. Always, we should not be doing intense workouts as they are not good for the body.

There main aim is to burn as much as calories within the limited amount of time that they can spend.

Problem statement:

Given 2 kinds of workouts -less intense and high intense workouts, the goal is to order the given exercises for 'n' days such that the constraint is satisfied which is : a person doing high intensity workout should be taking the previous day as a leave.

By satisfying the constraint, the goal of our project is to maximise the number of calories a person can burn in 'n' days.

Objective:

The main objective of the project is to maximise the number of calories burnt in a given set of days.

DAYS	1	2	3	4	n
LOW CALORIE	L1	L2	L3	L4	Ln
HIGH CALORIE	H1	H2	H3	H4	Hn

Description:

- Firstly , the person can choose the exercises from the list. All the high calorie ones and low calorie ones will be placed in 2 separate arrays.
- Using dynamic programming we arrive at a recurrence [1]

$$OPT(0) = 0$$

$$OPT(1) = \max(L1, H1)$$

$$OPT(i) = \text{MAX} \left\{ \begin{array}{l} L_i + OPT(i-1) \\ H_i + OPT(i-2). \end{array} \right.$$

- After finding all the values for the days, the results are displayed as to which workout to be done on which day and the maximum calories are printed as the result.

Dynamic programming :

Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial. For example, if we write simple recursive solution for fibonacci numbers , we get exponential time complexity and if we optimize it by storing solutions of subproblems, time complexity reduces to linear.

Applications: in gyms or in personal training devices.

4. Design:

4.1 Strategy:

Dynamic programming is used in order to maximize the solution.[2] optimization over plain recursion is used.

The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

Steps:

Step 1: A Recurrence Consider different ways of splitting up or restricting the overall problem into subproblems or subcases, and come up with a recurrence

Step2:

Step 2: Induction Prove the recurrence correct by induction.

Step 3: Memoize If there are overlapping subproblems, memoize.

4.2 Algorithm:

```
def max_cal(low, high):  
    n = len(low)  
  
    # Make low and high 1-indexed by adding dummy values to the  
    # beginning. This works much more nicely since we want opt[i] =  
    # maximum for working i weeks, and opt[0] = 0 is a natural base  
    # case, so we want to have the weeks be 1-indexed. (Adding extra  
    # "padding" elements to arrays to make room for base cases is a  
    # common technique with DP problems.)  
    low = [0] + low  
    high = [0] + high  
  
    # Initialize opt[0..n] with all zeros.  
    opt = [0] * (n+1)  
  
    # The first base case, opt[0] = 0, is already taken care of.  
    # For opt[1], just take the higher of the two jobs.  
    opt[1] = max(low[1], high[1])  
  
    # Now loop through remaining weeks.  
    for i in range(2, n+1):  
        # How much could we make taking the low or high stress job?  
        low_total = low[i] + opt[i-1]  
        high_total = high[i] + opt[i-2]  
  
        # The optimal for weeks 1..i is the higher of the two  
        opt[i] = max(low_total, high_total)  
  
    return opt[n];
```

5.C Language Implementation:

```
#include<stdio.h>
#include<stdbool.h>
int max(int x, int y)
{
    if(x>y)return x;
    else return y;
}

void tasks (int low[],int high[],int n)
{ int i;int low_tot,high_tot;
  bool take_high[n+1];
  int opt[n+1];
  int cal[n+1];
  for(i=0;i<=n;i++)cal[i]=0;
  for(i=0;i<=n;i++)
  {
      take_high[i]=false;
      opt[i]=0;
  }
  opt[1]=max(low[1],high[1]);
  if(high[1]>low[1])
  { take_high[1]=true;
  }
  else
```

```

{
    take_high[1]=false;
}

for(i=2;i<=n;i++)
{
    low_tot=low[i]+opt[i-1];
    high_tot=high[i]+opt[i-2];
    opt[i]=max(low_tot,high_tot);
    if(low_tot<high_tot)take_high[i]=true;
    else
        take_high[i]=false;
}

int w=n;
char schedule[n];
while(w>0){ if(take_high[w]==true)
    {
        schedule[w]='H';
        cal[w]=high[w];
        w=w-2;
    }
    else
    {
        schedule[w]='L';
        cal[w]=low[w];
        w=w-1;
    }
}

printf("\n***** the activities are:*****\n\n");

```



```

    for(i=1;i<=n;i++)
    {
        if(schedule[i]=='H'||schedule[i]=='L')
            printf("%c ",schedule[i]);
        else printf("---- ");
    }
    printf("\n\n the calories are\n ");
    for(i=1;i<=n;i++)
    {
        printf("%d ",cal[i]);
    }
    printf("\n\n HOORAY!!!the maximum calories you can burn is %d ",opt[n]);
    printf("\n\n");
}

```

```

int main()
{
    printf("*****WELCOME*****");
    printf("\n\n");
    printf("\n\n The gym is offering you the following packages \n" );
    printf("\t High calorie burning exercises \n:");
    printf("*****");
    printf("\n JUMPING ROPE    : 750 cal/hr");[3]
    printf("\n STAIR SPRINTS   :800 cal/hr");
    printf("\n KICK BOXING      : 560 cal/hr");
    printf("\n CYCLING INTERVALS : 570 cal/hr");
    printf("\n RUNNING           : 600cal/hr");
    printf("\n KETTLEBELL CIRCUIT : 700 cal/hr");
    printf("\n STATIONERY BIKE    : 500 cal/hr");
}

```

```

printf("\n*****");
printf("\n\n Low calorie burning exercises :");
printf("\n ROWING MACHINE           :481 cal/hr");
printf("\nSTAIRS                     :600 cal/hr");
printf("\nSTRENGTH TRAINING             :550cal/hr");
printf("\nMETABOLIC RESISTANCE TRAINING CIRCUIT :505 cal/hr");
printf("\nELLIPTICAL                     :450 cal/hr");
printf("\nBATTLE ROPES                   :555 cal/hr");
printf("\nPOWER YOGA                     :665 cal/2hrs");
printf("\nmeditation and simple yogaasans   :100 cal/hr");
printf("\t*****");
int lowcal[8]={481,600,550,505,450,555,665,100};
int highcal[7]={750,800,560,570,600,700,500};
int n,i,low[10],high[10],exer,j,flag=0;
low[0]=0;high[0]=0;
printf(" \n\n Please Enter the number of days!!!:");
scanf("%d",&n);
if(n<=0)
{
    printf("please enter 1 or more than 1 day(s)!! Try next time\n\n\n");
    return 0;
}
printf(" enter the %d low calorie exercises you chose :\n",n);
for(i=1;i<=n;i++)
{ while(1)
    {
        scanf("%d",&exer);
        for(j=0;j<8;j++)
        { if(exer==lowcal[j])
            { flag=1; break; }
        }
    }
}

```

```

        else flag=0;
    }
    if(flag==1)
    {low[i]=exer;
    break;
    }
    else printf("please enter valid exercise: ");
}

}

flag=0;
printf(" enter the %d high calorie exercises you chose:\n",n);
for(i=1;i<=n;i++)
{
    while(1)
    {
        scanf("%d",&exer);
        for(j=0;j<7;j++)
        { if(exer==highcal[j])
            {flag=1; break;}
            else flag=0;
        }
        if(flag==1)
        {high[i]=exer;
        break;
        }
        else printf("please enter valid exercise: ");
    }
}
}

```

```

printf("\n*****THANKYOU*****");

tasks(low,high,n);

}

```

6.OUTPUTS/Results :

```

C:\Users\boddi\Documents\daa\bin\Debug\daa.exe
RUNNING : 600cal/hr
KETTLEBELL CIRCUIT : 700 cal/hr
STATIONERY BIKE : 500 cal/hr
*****

Low calorie burning exercises :
ROWING MACHINE :481 cal/hr
STAIRS :600 cal/hr
STRENGTH TRAINING :550cal/hr
METABOLIC RESISTANCE TRAINING CIRCUIT :505 cal/hr
ELLIPTICAL :450 cal/hr
BATTLE ROPES :555 cal/hr
POWER YOGA :665 cal/2hrs *****

Enter the number of days:5
enter the 5 low calorie exercises you chose :
505 665 450 665 665
enter the 5 high calorie exercises you chose:
500 500 500 500 500

*****THANKYOU*****
***** the activities are:*****

H L L L L

the calories are
500 665 450 665 665
the maximum calories you can burn is 3005
Process returned 0 (0x0) execution time : 26.646 s
Press any key to continue.

```

o/p 1.Output when all the high intensity workouts have same no.of calories

```
C:\Users\boddi\Documents\daa\bin\Debug\daa.exe
KICK BOXING      : 560 cal/hr
CYCLING INTERVALS : 570 cal/hr
RUNNING          : 600cal/hr
KETTLEBELL CIRCUIT : 700 cal/hr
STATIONERY BIKE  : 500 cal/hr
*****

Low calorie burning exercises :
ROWING MACHINE      :481 cal/hr
STAIRS              :600 cal/hr
STRENGTH TRAINING   :550cal/hr
METABOLIC RESISTANCE TRAINING CIRCUIT :505 cal/hr
ELLIPTICAL          :450 cal/hr
BATTLE ROPES        :555 cal/hr
POWER YOGA          :665 cal/2hrs *****

Enter the number of days:0
enter the 0 low calorie exercises you chose :
enter the 0 high calorie exercises you chose:

*****THANKYOU*****
***** the activities are:*****

the calories are

the maximum calories you can burn is 0
Process returned 0 (0x0)   execution time : 4.235 s
Press any key to continue.
```

o/p 2. Output when 0 days are given as i/p.

```
C:\Users\boddi\Documents\daa\bin\Debug\daa.exe
RUNNING          : 600cal/hr
KETTLEBELL CIRCUIT : 700 cal/hr
STATIONERY BIKE  : 500 cal/hr
*****

Low calorie burning exercises :
ROWING MACHINE      :481 cal/hr
STAIRS              :600 cal/hr
STRENGTH TRAINING   :550cal/hr
METABOLIC RESISTANCE TRAINING CIRCUIT :505 cal/hr
ELLIPTICAL          :450 cal/hr
BATTLE ROPES        :555 cal/hr
POWER YOGA          :665 cal/2hrs *****

Enter the number of days:1
enter the 1 low calorie exercises you chose :
665
enter the 1 high calorie exercises you chose:
800

*****THANKYOU*****
***** the activities are:*****

H

the calories are
800
the maximum calories you can burn is 800
Process returned 0 (0x0)   execution time : 15.720 s
Press any key to continue.
```

o/p 3. Output when a single day is the input

7.Test cases:

Test case-1.

->If an exercise not in the given menu is chosen , the user is directed to again enter the valid exercise

```
C:\Users\boddi\Documents\daa\bin\Debug\daa.exe
CYCLING INTERVALS : 570 cal/hr
RUNNING           : 600cal/hr
KETTLEBELL CIRCUIT : 700 cal/hr
STATIONERY BIKE    : 500 cal/hr
*****

Low calorie burning exercises :
ROWING MACHINE      :481 cal/hr
STAIRS              :600 cal/hr
STRENGTH TRAINING   :550cal/hr
METABOLIC RESISTANCE TRAINING CIRCUIT :505 cal/hr
ELLIPTICAL          :450 cal/hr
BATTLE ROPES        :555 cal/hr
POWER YOGA          :665 cal/2hrs *****

Enter the number of days:4
enter the 4 low calorie exercises you chose :
678
please enter valid exercise: 505
300
please enter valid exercise: 505
505
505
enter the 4 high calorie exercises you chose:
500
4444
please enter valid exercise: 800
800
800

*****THANKYOU*****
***** the activities are:*****

L L L L

the calories are
505 505 505 505
the maximum calories you can burn is 2020

Process returned 0 (0x0)   execution time : 50.062 s
Press any key to continue.
```

Test case-2:

When a person has to take a leave following which he needs to do a high intensity workout.

```
C:\Users\boddi\Documents\daa\bin\Debug\daa.exe
STAIR SPRINTS : 800 cal/hr
KICK BOXING : 560 cal/hr
CYCLING INTERVALS : 570 cal/hr
RUNNING : 600cal/hr
KETTLEBELL CIRCUIT : 700 cal/hr
STATIONERY BIKE : 500 cal/hr
*****

Low calorie burning exercises :
ROWING MACHINE : 481 cal/hr
STAIRS : 600 cal/hr
STRENGTH TRAINING : 550cal/hr
METABOLIC RESISTANCE TRAINING CIRCUIT : 505 cal/hr
ELLIPTICAL : 450 cal/hr
BATTLE ROPES : 555 cal/hr
POWER YOGA : 665 cal/2hrs
meditation and simple yogaasans : 100 cal/hr *****

Please Enter the number of days!!!:3
enter the 3 low calorie exercises you chose :
600
100
100
enter the 3 high calorie exercises you chose:
7900
please enter valid exercise: 800
800
800

*****THANKYOU*****
***** the activities are:*****

H ----- H

the calories are
800 0 800

HOORAY!!!the maximum calories you can burn is 1600

Process returned 0 (0x0) execution time : 29.523 s
Press any key to continue.
```

Test case 3:when negative number of days are given as input, program sends an error message

```
C:\Users\boddi\Documents\daa\bin\Debug\daa.exe
High calorie burning exercises
*****
JUMPING ROPE : 750 cal/hr
STAIR SPRINTS : 800 cal/hr
KICK BOXING : 560 cal/hr
CYCLING INTERVALS : 570 cal/hr
RUNNING : 600cal/hr
KETTLEBELL CIRCUIT : 700 cal/hr
STATIONERY BIKE : 500 cal/hr
*****

Low calorie burning exercises :
ROWING MACHINE : 481 cal/hr
STAIRS : 600 cal/hr
STRENGTH TRAINING : 550cal/hr
METABOLIC RESISTANCE TRAINING CIRCUIT : 505 cal/hr
ELLIPTICAL : 450 cal/hr
BATTLE ROPES : 555 cal/hr
POWER YOGA : 665 cal/2hrs
meditation and simple yogaasans : 100 cal/hr *****

Please Enter the number of days!!!:-22
please enter 1 or more than 1 day(s)!! Try next time

Process returned 0 (0x0) execution time : 3.147 s
Press any key to continue.
```

8.Conclusion:

To conclude, for people looking for maximum benefit in their workout our project provides a major help for them to organise their workouts so that at the end of the session they have gained maximum calories with minimum effort and gaps between 2 intense workouts.

9.References :

- [1]Algorithm Design text book :Jon Klienberg and Eva tardos
- [2]<http://ozark.hendrix.edu/~yorgey/382/static/dynamic-programming.pdf>
- [3]<https://finance.yahoo.com/news/14-best-calorie-burning-exercises-120000954.html>