

CSE213: Data Structures and Algorithms

Year-Semester: 2019-20 Even Semester

LNMIT, Jaipur

Instructions

1. The project is worth 10% of the course marks.
2. Most desired size of a team is 7 students. Only in very special situations an instructor may permit a team with 5 or 6 students. All teams will develop the projects based on the same specifications.
3. A team may be made of students from different sections of CSE213: DSA. Each team will be assigned a supervising instructor.
4. The projects will be marked using multiple assessments during the semester. These assessments can be divided into two main groups based on the criteria used for marking.
5. First group of assessments is based on the progress made by the team. These assessments are based on the milestones. Three milestones have been set for this project. Total worth of these three milestone assessments is 6 marks (out of total 10 marks for the project).
6. For full credit for a milestone, the team must submit the code by its set due date/time. Late submission of a milestone will have lower assessed score for the team. The submitted programs will be demonstrated by the teams to their supervising instructor for marking.
7. The second group of assessments are based on a series of competitions among the teams. A DSA Lab week, near the end of the semester, has been designated for the project competitions. The competition stage will be used to find the correctly implemented projects as well as those that perform better than the projects from the other teams. The competition-based assessment is worth 4 marks.

Overview of the Project

Our project specifications have been derived from a class assignment given at MIT for a first-year programming and problem-solving course. In this project student teams are invited to write a computer program for a word game similar to the game of scrabble (<https://en.wikipedia.org/wiki/Scrabble>). At Rupa ki Nangal, project teams will implement the game in C using the DSA (and CP) lessons. In addition, there will be topics and skills that the teams need to learn beyond what is covered in their classes or labs.

The development efforts and progress achievements will be marked equally for each member of the team and will be based on the milestones to be reached by the set dates and times. This part is worth 6 marks. This assessment has already been described previously in the introductory section above.

The inter-project competition is worth 4 marks for the teams. However, we recognise that not all team members may contribute equally to these efforts. Suppose a team with m students, scores n marks through competitions. The team will get $n * m$ marks to distribute among its members.

This is so because it is possible that competitive development would require special additional efforts and not all team members may be contributing equally to this goal. Only the members of the team know the contributions made and time devoted by the active mates.

The course instructors would accept team decisions to allocate these marks to the team members. In case of a disagreement within the team, each member will be assigned the same mark ($n - 0.5$).

Project Specifications

The purpose of the game program is to form a word using a bag of letters given as input. Not every given letter in the bag is distinct; some letters may be present more than once.

The project program forms a word. There are rules to compute the score for deciding a winning program.

In what follows we provide the details of this game which is like a popular game of Scrabble.

A dictionary of words has been downloaded from the Internet. This is a text file of about 1.7 million words. The dictionary has been placed in DSA-Lab classroom.google.com site with code 6vw7dlh.

The word constructed by a team should be made from the letters given to the team and it should be a word in the dictionary. (The word will be constructed by the programs that the teams develop in this project exercise. Please note, in this document/description, we may use the term *team* to mean the program developed by the team for this project.)

Each letter in the English alphabet is assigned a point value. We use the scores used by Scrabble. This is a fixed mapping and is set in the table below. Upper-case and lower-case letters are treated as the same.

Point value	Letters
0 points	* (An unspecific letter that matches any vowel)
1 point	E, A, I, O, N, R, T, L, S, U
2 points	D, G
3 points	B, C, M, P
4 points	F, H, V, W, Y
5 points	K
8 points	J, X
10 points	Q, Z

To play the game each team will get a bag of letters (one or more letters may repeat any number of times in a given bag). Each occurrence of a letter in the bag can be used only once to form a word.

In addition to 26 letters in the alphabet there is a special letter '*' – this special letter can match any vowel. For example, a bag of letters may be a, c, e, o, h, a, a, b. Some possible words one can construct are: ace, echo, cab, hoe.

The program your team develops will create a word of 3 or more letters from the dictionary.

How to Compute a Score?

The team score is computed as follows (Number of letters in the bag including * is s):

1. Compute the word weight `weight`. This is sum of the point values for the letters in the word formed by the program.
2. Compute multiplier `mult`. This value is computed as difference of two values `good` and `bad`. That is,
`mult = good - bad;`
 The rules for computing values `good` and `bad` are described below.
3. Value `good` is computed by assignment statement:
`good = (s - 1) * word_length;`
4. Minimum possible value for `bad` is 1 and it is computed as :
`bad = max((3 * (s + 1 - word_length)), 1);`
5. The following assignment computes the score for the word:
`score = weight * mult;`

It is obvious that for best score the team must find a long word with large weight. The strategy to find a good word is simple. Select several candidate words from the dictionary and then choose the one that gives the largest score.

It may not be possible to find all words that the given bag of letters supports as this work may take a long time. For competition stage, we set 2 minutes as the maximum time for your program to find a word. (*This may be changed later to keep the competition stage challenging*).

We will describe the milestones later. The milestones specifications will guide you in planning and developing the word-finding program and other support software. Next we describe the competition phase.

Competition

As stated previously, we do not expect all teams to be able to construct long words. The teams will compete over several stages in a DSA-Lab week marked for this purpose.

All teams will participate in the initial competition where up to 6 letters will be given to play the game. Several games will be played, and winners of the games noted. The participating teams will be divided into three bands based on their win records in this initial round of the competition. Following this teams will be awarded competition-based marks for this round as follows.

Based on the number of times a team has delivered top scoring word, the teams will be divided into three bands of about the same number of teams. This will be done on each lab session basis for this round. Teams in the top band will be assigned 1 mark. The other teams would get 0.5 mark if they achieved a top score in one game. The teams in the bottom band will not continue to the next round of the competition.

The next round of competition will be played with bags of 7 letters on the same basis as the first round. Few more teams will drop out of the competition.

Final two rounds of the competition will be among all teams in the class. The bags of letters used will be of 9 and 11 letters. Again, about a third of the teams will be eliminated at the end of the first round of full class competition.

We have already explained that the teams will be able to distribute marks earned in the competition round to its members and advise the instructors to record these marks.

Milestones and Submissions

Three milestones have been formed to help the teams develop their projects in a systematic way and progress to complete the project on time.

Each milestone is worth 2 marks. A milestone reached by the set date will be tested by the team's supervising instructor and marked. The team that is late to complete the milestone will receive only 1 mark if they complete it within a week after the set date/time.

There is no credit if the milestone is delayed more than a week. However, it is crucial and necessary to complete the milestone as other parts of the project depend on the specified functionality for the milestone.

Milestones are also important and helpful as they provide an opportunity to test your code and make sure that the necessary functions are running well and correctly before you write code that uses these functions.

Milestone 1: Code to Compute Score

(Due date: 15 Feb 2020 11:59PM)

The project program should be submitted on DSA-Lab classroom by the set deadline. It will be discussed and assigned marks by the team's supervising instructor in a lab session that is convenient to the team and the instructor.

The tasks to be completed for this milestone are:

1. Design and implement a suitable user interface to receive letters in *a bag of letters*.
2. Design and implement an interface to receive a word.
3. Test that the word is made of the letters in the bag and it is a word in the dictionary. Report if the word is not valid.
4. Compute and report the score for the word.

Please keep a copy of this program for use later. During the competition phase, this program will be useful to test the claims of words and scores other teams may make. You do not wish to lose a game if your competitor makes a false claim at a competition.

Milestone 2: Is a Word Possible from the Letters in the Bag?

(Due date: 11 March 2020 11:59PM)

For a given bag of letters, we need a quick and easy way to search matching words from the dictionary. Every letters in the word must be in the bag. If the bag does not have a letter in the word then the word cannot be made from the letters in the bag.

From these requirements it is easy to see that we need to pre-arrange the dictionary and words in it before the start of a game. Once a bag of letters is given we need to quickly identify several promising words that we would test for validity and for the best score.

The first step of collecting promising words is a search. The search is best organised so that program spend only a small effort to find many promising words – we may not have time for

identifying all words that can be formed. The teams who organise this part well will be the winners in the competition round.

What is a good organisation for the dictionary? Here are some hints and suggestions. You may have even better ways.

1. A bag of 8 letters will not create a 9-letter (or larger) word. So split the dictionary into a number of files each with words of some fixed size(s) only. Search for a word will be limited to only the files with words of right sizes.
2. Words in the dictionary could be searched based on the weight. Let each sub-dictionary file be sorted by the points. If the search is done sequentially, the most promising words should be found first. Lighter words could be placed later in the file.
3. You may even plan for each sub-dictionary to be split into smaller files based on the high-point value letters the words contain. For example, words of length 5 with letter Q may be a sub-dictionary. One will search this file only if letter Q is in the bag. The file can be ignored if letter Q is not in the given bag.

These are just a few hints and there is no best answer for the best organisation of the dictionary. Fortunately, this issue is for implementation in Milestone 3.

The task for the teams to demonstrate for Milestone 2 is described next. This will require some further reading about C programming.

Given a bag of letters and a word, we want a quick way to see if the word can be constructed from the letters in the bag. For this we may construct a signature for each word.

A useful signature is a bit string of 26 bits. For each letter in the alphabet we designate a bit to mark letter's presence in the word. A 32-bits `int` value is a useful container for a word signature. Let us call this signature `wordSign`.

Also use the letters in a bag of letters to create a signature of the available letters. Let us call it `bagSign`. If letters in a bag do not have all letters in a word, `wordSign` will have a set bit that is not set in `bagSign`. This is a quick way to reject unhelpful words.

Students will need to learn appropriate C operators for Boolean operations on bit strings.

Example:

Suppose we have words CHOOSE and MOUSE. The bag of letters comprise A, E, O, U, C, C, H and S.

Word/Bag	Signature (26-bits)	Comments
CHOOSE	00101001000000000010000000	Every set (1) bit has a set bit in <code>bagSign</code>
Bag of letters	10101001000000100010100000	7 set bits. C occurs two times in bag.
MOUSE	00001000000010100010100000	Some set bits are not set in <code>bagSign</code>

Milestone 3: Construct and Announce the Best Word

(Due date: 05 April 2020 11:59PM)

In the final milestone of the project, teams will get a bag of letters to create a word that delivers best score to the team. This word can be found by first creating a pool of many promising words of some fixed size. Remember there is a time constraint. This means, teams cannot create a pool of all words!

Size of the pool will depend on the speed of the program the teams have developed. Faster programs can create a bigger pool. Slow programs must restrict the size of the pool to fewer words.

Once the pool is created, find the best scoring word in the pool as the word to announce.

Did the team miss to include a good word in the pool? Use function `time()` in library `time.h` to determine the available time that remains for the team. If there is enough time remaining, make a second pool of additional words. And, see if a word can give better score than the one found previously.

Your program may do yet more round(s) of word searching if remaining time on your program clock permits.

You have nearly completed your project now. Finalise the program and submit it.

You may now work on your program to be ready for the competition round. Good luck.

Milestone Test Cases and Competition

To encourage all teams to successfully complete the milestones, tests used for milestone submissions will have bags with only 4 to 6 letters. The students can decide to use smaller dictionary with fewer words consistent with this assurance on the size of bag of letters.

Teams may use a more liberal time constraint of 3 minutes. Also, teams are free from the best scoring word obligation when being assessed for the milestones.

The participant teams at the competitions must note that all teams are required to use a computer in a CSE lab. These computers may be slower than the personal computers/laptops the students own. Also, the teams need to program the clock to display the time taken by their program to print the word.

After the Competition

After completion of the competition round, all submitted programs will be opened to the class. Students can learn from what their mates did.

Also, this will give an opportunity to others to report cheating cases. For example, if two teams have the same code! Or, code submitted for the project was copied from some website.

The cheating team will be punished with a big penalty score. The reporting student/team will get some additional marks added to them.