Day 1 work summation

01 October 2020 20:17

I am in search of algorithms on which we can built our story . I started searching for the algorithms which can be related to real life in a simple and precise manner. Amid the search "THE APPROXIMATION ALGORITHMS" struck my mind, which I came to know when I am exploring for project idea, and as the name sounds like it is related to our daily life. I took some articles from senior, went through some lectures of approximation algorithms. After some introduction to approximation algorithms the word "NP complete problem" started to come over again and again . For almost an hour I have gone through the concepts of NP complete concepts (including P complete problems).

Today I made some research about approximation algorithms which would help our project to move forward.

Day 2 work - 3 hrs

Monday, October 5, 2020 9:40 PM

Today I am happy to learn some new concepts and algorithms. I have learned the concepts of NP class of problems, the difficulty in dealing such problems in real life. I have understood the notion of NP completeness, NP hard etc. I have understood the notion of P=NP. I have gone through these concepts as I was trying to get some algorithms which would be suitable to inscribe in out project. The topic approximation (which we were trying to inscribe in the story) needs the knowledge of NP hard and NP completeness classes. I read about the VERTEX COVER
Problem . Although it is NP HARD problem there is a good dynamic programming coupled with binary search solution . I felt the way the algorithm was built interesting. As it seems that it can be used in many real situations . Eg: Board meeting, inviting delegates from different organizations etc. I studied the Travelling sailsman problem . I thought it is simple dynamic programming but it screwed me up for some time. I learned that the division of a BIG problem into small problems needs a very precise proof that the smaller problems can combine together to form the solution of a bigger problem. Although both the algorithms have exponential time complexity they are better than brute force solution. I have discussed with the writers of the project about the algorithms .

Vertex Cover – 4.5 hrs

Friday, October 30, 2020 9:11 PM

Discussed the generalised version of the vertex cover algorithm(applicable to all type of graphs) with teammates (Trinadh) and have done work on the special cases of the vertex cove solution.

I have researched on Vertex cover problem in trees an d bipartitie graphs. In trees the vertex cover problem can be done in polynomial time. I understood and analyzed the algorithms of vertex cover problems in trees. I also researched upon the vertex cover algorithms since their properties makes it easy to find the vertex cover problem. I have learned the max flow problem which helps us to do our problem. I have gone through many google scholar articles and completed a week of a coursera course (Shortest Paths Revisited, NP-Complete Problems and What to do about them). We are thinking for an idea having this algo. I took around 5 hours to research and lot more time to analyze the content. We are also considering the proofs of the algorithms so that we can get the bring in understanding of the algorithm.

Sailsman Problem – 4 hours

Friday, October 30, 2020 9:59 PM

The travelling Sailsman problem seems to be much relevant to the daily life situations. The travelling salesman problem is described as below:

" A person has to reach every node of the graph structure travelling minimum distance along with extra constraint of reaching every node exactly once. "

The travelling salesman problem is NP-Complete problem. So there is no polynomial time algorithm. We have developed the following situation. We are trying to change the subproblems in the bellman ford algorithm to find the solution. We used subproblems like reaching a node j in a path of I nodes reaching every node exactly once. But that ran into the problem that we can't construct larger testcases from smaller ones. So after referring some internet material and coursera courses . We have found an O(n^2 . 2^n) solution for the salesman problem. We analyzed the complexity and went through the proof and process very meticulously. We are also trying to include the process of getting through the solution as that it is what we do when we face a problem. We developed an idea for the next from this algorithm. We have discussed the algorithm with my teammates.

Included proof and logic – 2 hrs

Wednesday, November 04, 2020 4:08 PM

Our project group discussed about the episode 2. I have written the explanation of logic and proof the algorithm. I have mentioned it under the heading of appendix. The approach begins with taking different sub-problems and checks whether they are good enough to solve the problem. I have described the approach to the solution and decided the subproblem which will lead us to solution. The solution follows dynamic programming, but we can't make it polynomial time since it is a NP- Complete problem. I have described the Optimal substructure of the problem and correctness proof. The correctness proof follows induction and recursion correctness is trivial. I have written the pseudo code for the problem. We have analyzed the time complexity of the algorithm.

Different Approach for Knapsack – 3.5hrs

Sunday, November 08, 2020 8:24 PM

I have worked on some heuristics for the upcoming episode. I have worked on knapsack problem. The Knapsack problem is a popular one. The knapsack problem can be related to the daily life in many situations. In knapsack problem we try to maximize the value with a certain weight constraint. There will situations that the values are less and weights are very high (e.g.: population, average area) there the normal dynamic program approach of complexity O(N.W) may not work since the weight constraint is very high. So I have an studied an alternative dynamic program approach for the knapsack problem with complexity O (n^2 . Vmax) where Vmax is the max value of the values of objects given. This approach is very useful when the value of W is very less than n.Vmax which is quite possible. I have worked on this approach of solving of knapsack problem.I have gone through all knapsack algorithms once.

Knapsack Heuristic - 1 - 2.5 hrs

Sunday, November 08, 2020 9:01 PM

I have worked on the heuristic of knapsack problem. The heuristic works in this way it declares the weight/value ratio as the “Cash”. We calculate the value of cash for every object. It sorts them and takes the values in ascending order of the cash until the weight of chosen objects stay less than W(Capacity). We also observe if there is single object with less weight less than W and has the value more than the value obtained by taking the objects in the above-mentioned way.

I have gone through the analysis of the algorithm. The heuristic gives us an solution which is always greater than 50% of the actual solution. If the actual solution is ‘a’ then the solution by above heuristic lies between ½ *a to a. In many situations it will be closer to the solution if n is large the distribution is well diverse. For this approach I have read up article “Heuristic can be good” in google. I have followed coursera material on this topic. I have researched about it and made a good understanding of the analysis of the algorithm.

Knapsack Heuristic –2 – 5 hrs

Sunday, November 08, 2020 9:02 PM

The previous heuristic has a guarantee that the solution is greater than the 50% of the optimal solution which in some cases may be inapplicable. I have a studied an dynamic programming heuristic. It has a parameter '€' (the inaccuracy measure) which deals with closeness of the solution to the optimal solution.

The heuristic uses the 2nd (mentioned previously , complexity : O((n^2).Vmax)) approach to the solve the heuristic. What we do in this heuristic is : We will divide the values of the objects with a particular number 'k'(based upon the accuracy need), and on the obtained results we will apply the second approach of knapsack problem(weights remain unchanged).

I have gone algorithm and proof of the algorithm(Proof of How can it guarantee the needed accuracy). The choice of value of k determines the accuracy obtained . Lower the value of k higher the accuracy. The analysis and proof has been taking too long for me to understand

Path retracing for the sails man travelling problem- 2.5 hrs

Thursday, November 12, 2020 4:14 PM

I have indulged in writing the pseudo code for the sails man travelling problem. The algorithm we have explained how to obtain the minimum cost for traversal without visiting any node twice. I have researched upon it . Have failed with logic and code many times. Discussed with my team mates. I have tweaked the dynamic programming logic with some additional structures so that I can remember the path and get the back when needed. I have added a function retrace_optimal_path it will retrace the path that has to be visited so that no node visited more than once. We have included that in the episode. I took more than 5 hours to complete the pseudo code.

APPENDIX - 2Hours

Thursday, November 12, 2020 4:34 PM

I have written the appendix given in the episode . I have included the explanation for the logic and the way the problem is solved . I have explained the algorithm , described the proof of correctness , and the pseudo code. I have included the subproblems and way the subproblems are exploited to obtain the solution of the problem. I have the included the optimal substructure(The way the subproblems lead to the solution). I have added the time complexity to the appendix. I have also discussed why our initial ways failed to lead us to the solution. I have included the pseudo code for the problem.

2-SAT problem – 3.25 hrs

Thursday, November 12, 2020 4:41 PM

For development of Episode – 3 I have worked upon the SAT problems and got to know that SAT problems are NP-complete and hence for the sake of simplicity(for story development and to get situations better) I have chosen to work on 2-sat problem. I have researched upon the 2-sat problem, read many articles on the 2-sat problems and gone through coursera material. I have learned the algorithm used for solving 2-sat problem. I have gone through the proof of correctness.

The 2-sat problem is basically the problem of determining if a Boolean formula is satisfiable or unsatisfiable. The algorithm solves it by constructing a graph.

I have worked to design a real life situation using the 2-sat problem. Since the problem implicitly mean that every pair has to true. I thought of situations where every pair has to do their work in such a way that their pair will be true and it also doesn't take away any chances for other pairs to be true. I visualized a pretty old fashioned kidnap(kind of) situation. I have told my teammates about the algorithm. It took 4 and half hours for me to get hold of the algorithm, to get the situations where I can apply the algorithms and discuss about it.

Final Algorithm for Episode-3 - 2hrs

Wednesday, November 18, 2020 7:10 AM

We have a meet which lasted for about 2 hours about the algorithm of 3rd episode. We have prepared algorithms like 2-Sat ,Approximate knapsack, min cut, local search and few more. We have discussed about the situation in which we can put the algorithm in a very balanced manner. We have discussed the situations for all the algorithms and finally decided upon the Approximate Knapsack Problem since it brings a nice continuation to the story and also It is a different domain of algorithms that we haven't touched upon.

Reviewed Episode – 3 – 30 min

Tuesday, November 17, 2020 11:11 AM

Our team has completed the work of the episode-3. So I have went through the whole episode to let my team mates know if there any changes to the algorithm part or the dialogue part/.Our algo team proposed some changes. I have taken care of dialogues at the algorithm introduction part so that we will be clear about the choice of algorithms.

Unsuccessful search for a Grand algorithm – 3.5hours

Sunday, November 15, 2020 7:26 PM

Our team thought of having of grand(U know what I mean 😊) algorithm for the story. By grand we mean it has to be very fitting into story and the story that we can weave around the story has to very appealing. I have searched in many algorithm courses in coursera and Udemy. I have gone through some bit stream algorithms(Used mostly in multimedia systems) ,some memory management algorithms, some mathematics tutorials on which the algorithms are built(For prime-duality algorithm).
I have gone through some problems for which the solutions has to be space efficient (Just introduction to the problem). I have googled for many mathematical clarification and for the real life situations where the algorithm is used often. But at last I am not satisfied with real life situations where they can be used. I haven't decided on any algorithm to propose ☹ .

Story idea for Episode-3 – 4 hrs

Thursday, November 12, 2020 4:42 PM

At last, after a long discussion and research on algorithms we finally proposed the story idea of the episode-3. We (Me, Harsha, trinadh) has developed the story line on our proposed algorithm (Approximation algorithm of knapsack). The story is briefed in the progress diary of Trinadh. We have discussed many scenarios and tried to include many algorithms into the Story. We don't want to bring the algorithm unnaturally into story or to put the algorithm in a very expected way. We have tried our best to maintain the continuation of the story and indulge our algorithm in a apt way. We have discussed almost for 4 hours on the story line and the details in the story. We have explained it our teammates so they can write the dialogues and proceed with other ones. Being a part of the algorithm team, we have mentioned the meets so that the we can support the choice of our algorithm. We have designed the story line in such a way that we won't be using expected algorithms.

Heavy Hitter problem – Streaming algorithm

Tuesday, November 17, 2020 11:12 AM

We are searching for a different domain of algorithms for our final episode. I have been searching for past 2 days. I have thought the streaming algorithms would be better to give a try. I have worked on some basics of streaming algorithms. I have thought that since we are giving much importance to the time based algorithms I have thought that choosing a space efficient algorithm would be better and hopefully there are many space efficient algorithms in streaming algorithms since the space is been a issue for the heavy data passage routers.

I am working on the heavy hitters problem. The heavy hitter problem is an approximation algorithm. The heavy hitter problem includes a lot of mathematics (especially probability). The way that the heavy hitter problem is solved is very fascinating. I have been studying the building blocks of the algorithm. I am following some google articles and listening to some lectures. In the heavy hitter problem the major building block is constructing a data structures with minimal space requirements and it output the number of times an item has occurred in a stream at that point. We call that data structure as point query data structure. It outputs the approximate value of the frequency of an item until that point. The guarantees include lot of probability stuff. For that purpose I have to study Chebyshev's inequality, Markov inequality and Chernoff bounds.

Chebyshev's inequality ,Markov's inequality, Chernoff bounds – 3.25hours

Tuesday, November 17, 2020 11:12 AM

For a better understanding of proofs of guarantees(approximation guarantees) of heavy hitters problem I thought going learning the Chebyshev's , Markov's inequalities and Chernoff bounds.

I have read the portion of inequalities and bound in mit portal and listened to some of the lectures which I didn't understand.

Analysis of Heavy hitters problem – 5.5 hrs

Wednesday, November 18, 2020 7:01 AM

After going through the building blocks of the heavy hitters problem(point query data structure). I have studied how the building blocks are brought together to form the final algo. What we do is:

Since our point query data structure accuracy depends upon the distribution of the elements. We would divide the stream into sub stream (buckets). We map the elements from the stream into the buckets and there we use the our point query data structure(since the distribution are much distributed) the algorithm will be having much accuracy guarantees. To improve the probabilities what we do is we take logarithmic hash functions and we will take the median of those. We can use median of medians algorithm for that purpose. I have gone through the COMPLICATED proofs and accuracy guarantees of the algorithms (which took me a lot of time).

I have analyzed the algorithm It maintain a 2-D array of total size O(k.log(N)) I have gone through the analysis of number of buckets to be taken so that the accuracy guarantees and postulates we have taken to prove the Accuracy guarantees will be satisfied. This is the TOUGHEST algorithm I have ever gone through.

Discussed about our work with my team mates – 3.25 hours

Wednesday, November 18, 2020 7:00 AM

We have discussed about the algorithms bought up by the algo team. We have discussed about the viability of different algorithm into the story . We have decided situations for the algorithm without losing the connectivity in the story. At last we have decided to have another algorithm at hand because the algorithm that I have bought up looked somewhat complicated to be put up in the story.

SCHEDULING – Job Shop algorithm , (Searching – 1.5hr)

Wednesday, November 18, 2020 1:29 PM

Since the Heavy hitter problem is based upon on many assumptions and lots and lots of math is involved in it out team have thought of having another algorithm in hand so I have thought of do some other algorithm. I have thought of doing some other algorithm. The algorithm which can fit , which could be presented better and which are also somewhat tricky become difficult to find. The resources to get to know about the algorithms have become tough to find. That is why I took more than an hour to search for an algorithm with satisfy the above terms.

Finally I have decided to take a try on the scheduling algorithms. I have searched for some scheduling algorithms and the Job Shop algorithm which is done using the constraint programming popped up as better one to carryon since It doesn't have many prerequisites. I have started to explore on the Job Search algorithm and the constraint programming. I have learned that the to deal with the precedence constraints and the disjunctive constraints. I have gone through some graphs which explain the concepts of constraints.

But as I proceed further I have found that I have to study a lot of other concepts to carry on with the topics since we have come to end I have stopped in the middle (☹) . I will carryon with it after having a algorithm in hand for our next episode.

Children Grouping algorithm

Friday, November 20, 2020 2:21 PM

To be in safer side our team has decided to have some side – algorithm so that we won't be in a trouble if we can't include the algorithm that we have thought of. So I have chosen this children grouping algorithm. Usually the naïve solution for this algorithm would be an exponential algorithm by trying about every children in every group.
We have come up with a polynomial-time algorithm for the problem as opposed to the exponential type algorithm. We have proceeded by using segments. The algorithm will follow like this:

If we have a group of children, it consists of several children and several points on the line corresponding to this group and the fact that the age of any two children in the group differs by at most one means that there exists a segment of length one on this line that contains all those points.

Now the goal becomes to select the minimum possible number of segments of length one, such that those segments cover all the points. Then, if we have such segments, we can just take all the points from that segment in the same group, and any two children in the group who differ by, at most, one year.

The algorithm was chosen because the solution is not so simple as the question seems.

Appendix for episode 3 – 3 hours.

Friday, November 20, 2020 2:20 PM

I have written the appendix for episode – 3 which is based upon the Approximate knapsack algorithm. I have included why we have chosen a new approach for solving the knapsack problem and why we have to modify the new approach. I have mentioned why we have chosen the approximation algorithm . The approximation algorithm we took have guarantees and math is involved in it. I have included all the math needed to show the guarantees provided by the algorithm. I have included the way by which we choose our parameters to satisfy the needed accuracy constraints. I have included the running time complexity in the appendix. I have explained the various conditions and included a detailed explanation of the algorithm and added some pseudo code snippets.