

Practice Test AOA

Class: SE B - 50

Date: 06/05/2020

Time: 10 AM to 11AM

Enter Roll No :	50
------------------------	----

Q. 1	What is longest common subsequence problem. Find LCS for the following string. X = ACBAED and Y = ABCABE.
------	---

Q. 2	Explain how branch and bound strategy can be used in 15 puzzle problem.
------	---

Q.1.

The "longest common subsequence problem" is the problem of finding the longest subsequence common to all sequences in a problem.

It differs from longest common substring problem.

$X = \text{ACRAED}$

$Y = \text{ABCABE}$

$\text{LCS}[i, j] =$

$$\begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ 1 + \text{LCS}[i-1, j-1] & \text{if } p_i = q_j \\ \max(\text{LCS}[i, j-1], \text{LCS}[i-1, j]) & \text{if } p_i \neq q_j \end{cases}$$

 y_i

0	1	2	3	4	5	6
---	---	---	---	---	---	---

x_i	0	A	B	C	A	B	E
-------	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

1	A	0					
---	---	---	--	--	--	--	--

2	█	0					
---	---	---	--	--	--	--	--

3	B	0					
---	---	---	--	--	--	--	--

4	A	0					
---	---	---	--	--	--	--	--

5	E	0					
---	---	---	--	--	--	--	--

6	D	0					
---	---	---	--	--	--	--	--

Anney Thapar B-50

PAGE No.	
DATE	/ /

Computation for row 1

$$LCS[1,1] \Rightarrow 1 + LCS[0,0]$$

$$= 1 + 0 = 1$$

$$LCS[1,2] \Rightarrow \max(LCS[0,2], LCS[1,1]) \\ = \max(0, 1) = 1$$

$$LCS[1,3] \Rightarrow \max(LCS[0,3], LCS[1,2]) \\ = \max(0, 1) = 1$$

$$LCS[1,4] \Rightarrow 1 + LCS[0,3] \\ = 1 + 0 = 1$$

$$LCS[1,5] \Rightarrow \max(LCS[0,5], LCS[1,4]) \\ = \max(0, 1) = 1$$

$$LCS[1,6] \Rightarrow \max(LCS[0,6], LCS[1,5]) \\ = \max(0, 1) = 1$$

Likewise

Amey Balavur B-50

Page No.	
Date:	/ /

Find Lcs

S1: $i = 6, j = 6, x_i = D, y_j = E$

$x_i \neq y_j$ (and $LCS[i, j]$ is derived from $LCS[i-1, j-1]$) so move vertically

(LCS[i-1, j] is same)

Solution set $S = \{D\}$

S2: $i = 5, j = 6, x_i = B, y_j = F$

$x_i \neq y_j$ so add x_i to solⁿ set
and move diagonally back

(E. Solution set is now $\{B, F\}$)

S3: $i = 4, j = 5, x_i = A, y_j = R$

$x_i \neq y_j$ and $LCS[i, j]$ is derived from $LCS[i-1, j]$ so move in any direction

Solution set $S = \{E\}$

S4: $i = 3, j = 8, x_i = L, y_j = B$

$x_i \neq y_j$ so add x_i to solⁿ set.
and move diagonally back

Solution set = $\{E, B\}$

S5: $i = 2, j = 4, x_i = C, y_j = A$

$x_i = y_j$ add x_i to sol¹ set.
move horizontal

sol¹ set $S = \{E, B\}$

S6: $i = 2, j = 3, x_i = C, y_j = C$

$x_i = y_j$ add x_i to sol¹ set.
move diagonally back

sol¹ set $S = \{E, B, C\}$

S7: $i = 1, j = 2, x_i = A, y_j = B$

$x_i \neq y_j$
move horizontally

sol¹ set $S = \{E, B, C\}$

S8: $i = 1, j = 2, x_i = A, y_j = A$

move diagonally back

sol¹ set $\rightarrow \{E, B, C, A\}$

LCS of $x + y \rightarrow ACBE$

LCS = ACBE

Q2. Explain the Branch and Bound algorithm.

Ans: Branch and Bound algorithm.

Branch and Bound strategy in 15 puzzle problem

- Branch and bound builds the state-space tree and find the optimal solution quickly by pruning few of the tree branches which does not satisfy the bound

- Backtracking can be useful where some other optimization technique like greedy or dynamic programming fail.

- Such algorithms are typically slower than their counterparts. In the worst case, it may run in exponential time. but careful selection of bounds and branches makes an algorithm to run reasonably faster.

- In branch-and-bound, all the children of E nodes are generated before any other live node becomes E node.

- Branch-and-bound technique in which E node puts its children in queue is called FIFO branch-and-bound approach.

- And if E node puts its children in the stack then it is called LIFO branch-and-bound approach.

- Bounding function are a heuristic function.

- Heuristic function computes the node which maximizes the probability of better search or minimizes the probability of worst search.

- Used to solve optimization problems

- Nodes in tree may be explored in depth first or breadth first order.
- Next move is always towards better solution.
- Entire state space tree is searched in order to find optimal solution.
- Application:
 - (1) Travelling salesman problem.
- Branch and bound technique generates all the child nodes of E-node before another node becomes E-node.
- Let $c(x)$ represent the cost of answer node x . The aim is to find minimum cost answer subnode.
- Search strategies like FIFO, LIFO and LC search differ in terms of sequence in which they explore the node in state space tree.
- Branch and bound method employ either BFS and DFS search.
- During BFS, expanded nodes are kept in a queue whereas in DFS nodes are kept on the stack.
- Example: FIFO branch and bound approach

- In this problem, there are 15 tiles, which are numbered from 0-15.
- Initial and goal arrangement.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	6	8		5	6	7	8	9	10	11	12	13	14	15	16
9	10	7	11	9	10	11	12	13	14	15	16	13	14	15	16
13	14	15	16	13	14	15	16	13	14	15	16	13	14	15	16

(i) Initial state (ii) Goal state

- There is always an empty slot in initial state.

- In the initial state, moves are the moves in which the tile adjacent to Es are moved to either left, right, up or down.
- Each move (left, right, up, down) creates a new arrangement in a tile.
- These arrangements are called different states of the puzzle.
- The initial arrangement is called as initial state and goal arrangement is goal state.
- The state space tree for 15 puzzle is very large because there can be $16!$ different arrangements.
- In state space tree, the nodes are numbered as per the level.
- Each next move is generated based on empty slot position.

- Edges are labelling accordingly to the direction in which empty space moves,
 - The root node becomes E-node
 - The child nodes 2, 3, 4, and 5 of this E-node get separated.
 - Out of which node 4 becomes an E-node and gets generated
 - Out of which node 214 gets generated an E-node. For this node the five nodes 10, 11, 12 gets generated
 - Then the node 10 becomes the E-node for which child nodes 22 and 23 gets generated
 - We can decide which node becomes E-node based on estimation formula
 - Cost estimation formula
- $$C(a) = f(a) + 2g(a)$$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

up	right	down	left
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

cost $f(a) + h(a)$ after first moves

Amey Thakur, B - 5¹⁰

PAGE NO.	/ /
DATE	/ /

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	8010	8011	8012	8013	8014	8015	8016	8017	8018	8019	8020	8021	8022	8023	8024	8025	8026	8027	8028	8029	8030	8031	8032	8033	8034	8035	8036	8037	8038	8039	8040	8041	8042	8043	8044	8045	8046	8047	8048	8049	8050	8051	8052	8053	8054	8055	8056	8057	8058	8059	8060	8061	8062	8063	8064	8065	8066	8067	8068	8069	8070	8071	8072	8073	8074	8075	8076	8077	8078	8079	8080	8081	8082	8083	8084	8085	8086	8087	8088	8089	8090	8091	8092	8093	8094	8095	8096	8097	8098	8099	80100	80101	80102	80103	80104	80105	80106	80107	80108	80109	80110	80111	80112	80113	80114	80115	80116	80117	80118	80119	80120	80121	80122	80123	80124	80125	80126	80127	80128	80129	80130	80131	80132	80133	80134	80135	80136	80137	80138	80139	80140	80141	80142	80143	80144	80145	80146	80147	80148	80149	80150	80151	80152	80153	80154	80155	80156	80157	80158	80159	80160	80161	80162	80163	80164	80165	80166	80167	80168	80169	80170	80171	80172	80173	80174	80175	80176	80177	80178	80179	80180	80181	80182	80183	80184	80185	80186	80187	80188	80189	80190	80191	80192	80193	80194	80195	80196	80197	80198	80199	80200	80201	80202	80203	80204	80205	80206	80207	80208	80209	80210	80211	80212	80213	80214	80215	80216	80217	80218	80219	80220	80221	80222	80223	80224	80225	80226	80227	80228	80229	80230	80231	80232	80233	80234	80235	80236	80237	80238	80239	80240	80241	80242	80243	80244	80245	80246	80247	80248	80249	80250	80251	80252	80253	80254	80255	80256	80257	80258	80259	80260	80261	80262	80263	80264	80265	80266	80267	80268	80269	80270	80271	80272	80273	80274	80275	80276	80277	80278	80279	80280	80281	80282	80283	80284	80285	80286	80287	80288	80289	80290	80291	80292	80293	80294	80295	80296	80297	80298	80299	80300	80301	80302	80303	80304	80305	80306	80307	80308	80309	80310	80311	80312	80313	80314	80315	80316	80317	80318	80319	80320	80321	80322	80323	80324	80325	80326	80327	80328	80329	80330	80331	80332	80333	80334	80335	80336	80337	80338	80339	80340	80341	80342	80343	80344	80345	80346	80347	80348	80349	80350	80351	80352	80353	80354	80355	80356	80357	80358	80359	80360	80361	80362	80363	80364	80365	80366	80367	80368	80369	80370	80371	80372	80373	80374	80375	80376	80377	80378	80379	80380	80381	80382	80383	80384	80385	80386	80387	80388	80389	80390	80391	80392	80393	80394	80395	80396	80397	80398	80399	80400	80401	80402	80403	80404	80405	80406	80407	80408	80409	80410	80411	80412	80413	80414	80415	80416	80417	80418	80419	80420	80421	80422	80423	80424	80425	80426	80427	80428	80429	80430	80431	80432	80433	80434	80435	80436	80437	80438	80439	80440	80441	80442	80443	80444	80445	80446	80447	80448	80449	80450	80451	80452	80453	80454	80455	80456	80457	80458	80459	80460	80461	80462	80463	80464	80465	80466	80467	80468	80469	80470	80471	80472	80473	80474	80475	80476	804