Session Name: 25 - Review 3 5-5-2022 12-25 PM

Date Created: 5/5/22, 12:04:49 PM **Active Participants:** 93 of 111
Average Score: 6.14% **Questions:** 8

Results by Question

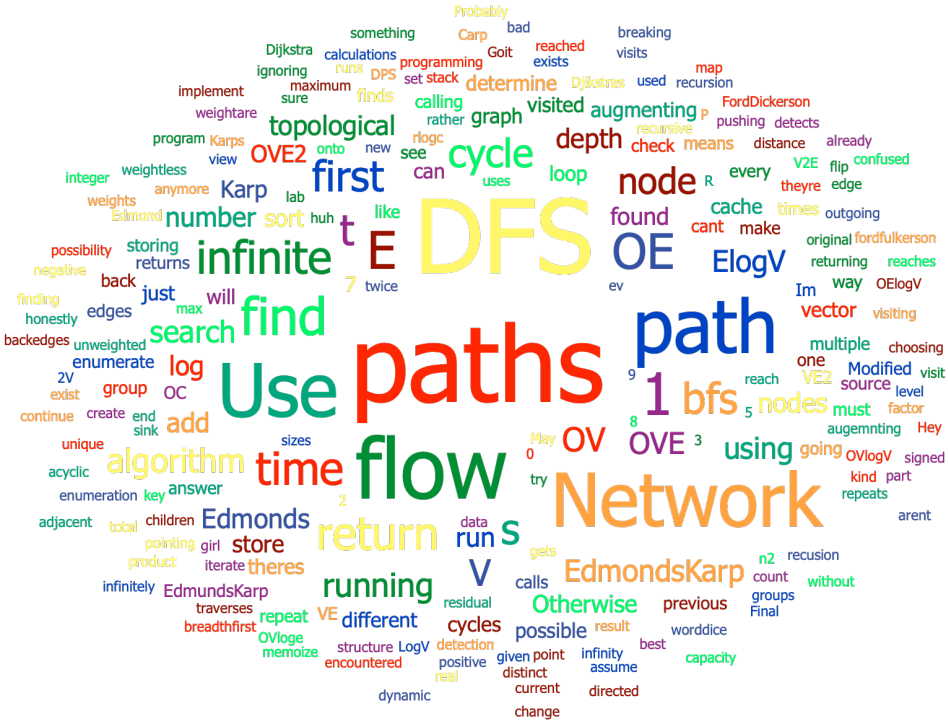
1. Please do question 1 which is on the screen in class. (Essay)

Responses
1
1) source to sink (network flow) 2) store paths in cache (a data structure or vector) 3) infinite if any distinct path repeats
2^V, O(E log V)
5
7
7
7
8, I would use network flow, O(E log V)
9 - Network flow, augmenting path with DFS - O(VlogV)
Because weights aren't given, assume edges have the same weight/are weightless. The best algorithm to find augmenting paths for an unweighted graph is Edmonds-Karp, which will be O(VE^2). This algorithm uses breadth-first search, running time O(E).
BFS multiple times using Edmond Karp's maximum flow algorithm- Each time it reaches t means there's a new path.
BFS, I'm honestly kind of confused,
DFS
DFS
Dfs
DFS
dfs
DFS
DFS - O(lVl)
DFS e+v
DFS in a while loop while storing used paths
Dfs O(v + e)
DFS O(V+E)

DFS storing previous paths. a loop means it is infinite. V + E
DFS V+E
DFS while choosing different nodes at each level, running time: O(E)
DFS with a stack and just iterate for each time t is reached. If the DFS detects a cycle, return -1. O(E) running time
Djikstras
Do a DPS without breaking after finding the first path, but rather returning 1. Add all the recursive calls up. If at any time the current node has been visited, return -1
dynamic programming to find all the paths that reach t from s. add up all the possible paths from outgoing edges from S using recursion. create a key of the nodes in a path and cache it to memoize. O(V)
E
E
E log V
Edmonds Carp O(E)
Edmonds Karp using calls of BFS. O(VE^2)
Edmonds-Karp, repeat until no more paths from s to t exist. O(V + E)
ElogV
ElogV
Enumerate DFS. -1 if visit same node in same enumeration. E log V
First run BFS to see if we can find node s from node s. If we can't then run Topological sort. O(E)
First, check for cycles using BFS O(V+E) visiting each nodes children. If there is a cycle return infinite paths. If no cycle use the edmonds-karp algorithm O(Elog(V)) to find all possible augmenting paths.
Ford-Dickerson and Edmunds-Karp (O(V + E))
ford-fulkerson
Goit to be real here. This is a bad answer. So, what I would do is run through DFS multiple times. For each run, I would factor in previous paths, and store those in a cache to check which way I am going and if the path already exists, we will try a different node to make a different path. The running time should be something like O(Vlog(e)).
Hey girl you must be a signed integer. If you just add me, I can change that point of view from negative to positive ;)
huh
i would use a topological sort
I would use network flow. ElogV. I store all the paths I make and if I repeat one it returns -1

as it will loop infinitely. Otherwise is just returns the total number of paths.
I would use the Edmonds-Karp network flow algorithm to find the path. It has a running time of O(VE^2)
I would use the Edmunds-Karp algorithm to find paths until there are no more paths through the graph which has a running time of O(C + R)
If there is a path going back to S, it has infinite paths. You can use depth first search calling it on s first and then calling it on each of its adjacent nodes.
If there is infinite paths then there is a node pointing back to s. Otherwise, use Edmonds-karp to find number of possible paths. O(VE^2)
If there was no possibility of having an infinite number of unique paths I would implement the Edmonds Karp algorithm (O(E)) to count the paths, otherwise theres no way to determine if there are an infinite number of paths as the program would not end
Infinite paths — return -1
It must be directed acyclic. May use DFS to find every path, store them as they're found. If more than n^2 paths are found, return -1.
I'm not sure
Modified Depth first search
Modified Dijkstra
Network flow
Network flow - E Log V
Network flow - find all augemnting paths with edmonds karp
Network Flow and Depth First Search: ElogV
Network Flow ElogV
Network flow O(e), find paths like worddice lab
Network flow using dfs for cycle detection, running time is O(V *E Log(V))
Network Flow with BFS.
Network flow, DFS, O(V+E)
Network Flow, O(E)
Network flow, rlogc
Network flow, while bfs finds a path, set the capacity of that path to 0 and flow to 1.
Network flow: once a path is found from s to t, flip original and residual, then continue until there is no path. Edmonds-Karp is VE^2
O(C)
O(E)
O(V+E)

Probably use topological sort to see in there is a cycle, if one then return -1, then use dfs to find all the paths from a to t. O(v + e) and O(e)
Run Bfs and network flow. V^2E
Topological sort. If there's a cycle, it's infinity, otherwise you group all the nodes into groups of the same topological distance. Your answer is the product of all group sizes. O(V+E)
Use a bfs or dfs o(v+e)
Use BFS with recusion. Return if t is visited, and there are infinite paths if a node visits s. ElogV
use depth first search until it can’t find a part anymore
Use depth first search, pushing path onto map each time. If cycle is encountered, return -1.
Use dfs and after each path if it gets you to t, add path to a vector. Final result is vector of paths to t
Use DFS to find each path and determine a cycle (for infinite paths). P (v + e)
Use Network Flow, which traverses every edge. If a node is visited twice, return -1
using DFS, after each path found, if it finds t, add backedges to path, if not, return -1
V*E
VE^2
We first use DFS on the source to determine if there are any cycles in the graph. DFS runs in |V| + |E|. If there is a cycle, there are an infinite number of paths, so we return -1. If there are no cycles, we use Edmonds Karp to enumerate all the paths, ignoring any max flow calculations



2. Please do question 2 which is on the screen in class. (Short Answer)

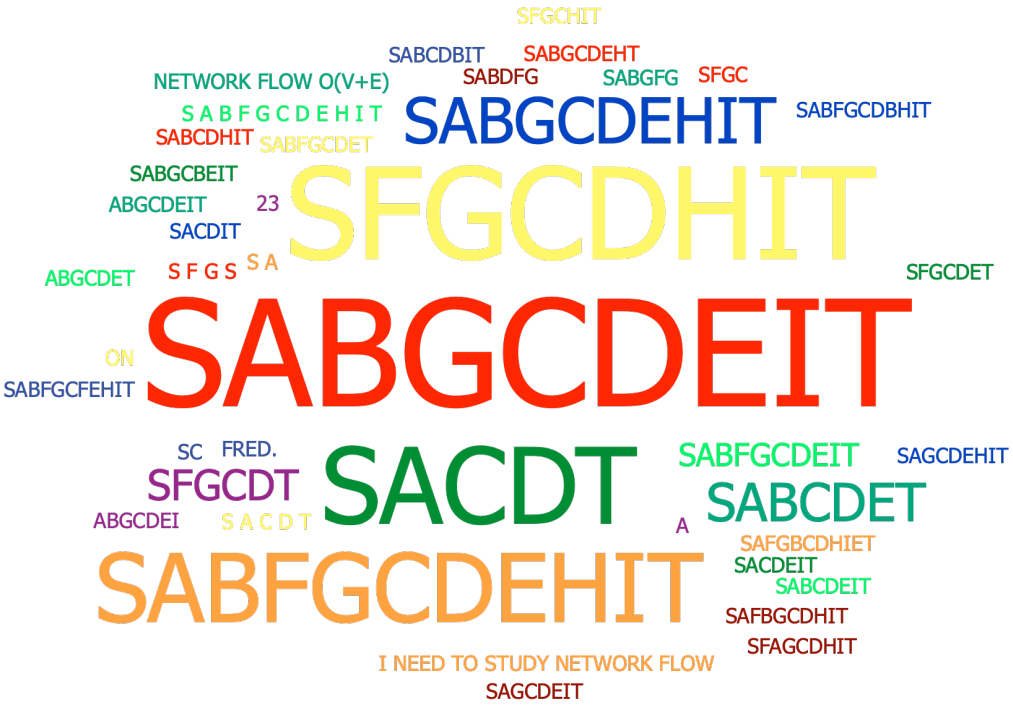
	Responses	
	Percent	**Count**
SABGCDEIT	14.61%	13
SFGCDHIT	12.36%	11
SACDT	10.11%	9
SABFGCDEHIT *(c)*	8.99%	8
SABGCDEHIT	5.62%	5
SABCDET	4.49%	4

SFGCDT	3.37%	3
SABFGCDEIT	2.25%	2
23	1.12%	1
A	1.12%	1
ABGCDEI	1.12%	1
ABGCDEIT	1.12%	1
ABGCDET	1.12%	1
FRED.	1.12%	1
I NEED TO STUDY NETWORK FLOW	1.12%	1
NETWORK FLOW O(V+E)	1.12%	1
ON	1.12%	1
S A	1.12%	1
S A B F G C D E H I T	1.12%	1
S A C D T	1.12%	1
S F G S	1.12%	1
SABCDBIT	1.12%	1
SABCDEIT	1.12%	1
SABCDHIT	1.12%	1
SABDFG	1.12%	1
SABFGCDBHIT	1.12%	1
SABFGCDET	1.12%	1
SABFGCFEHIT	1.12%	1
SABGCBEIT	1.12%	1
SABGCDEHT	1.12%	1
SABGFG	1.12%	1
SACDEIT	1.12%	1
SACDIT	1.12%	1
SAFBGCDHIT	1.12%	1
SAFGBCDHIET	1.12%	1
SAGCDEHIT	1.12%	1

SAGCDEIT	1.12%	1
SC	1.12%	1
SFAGCDHIT	1.12%	1
SFGC	1.12%	1
SFGCDET	1.12%	1
SFGCHIT	1.12%	1
Totals	**100%**	**89**

Keyword(s): SABFGCDEHIT

Keyword Matches: 8



3. Please do question 3 which is on the screen in class. (Short Answer)

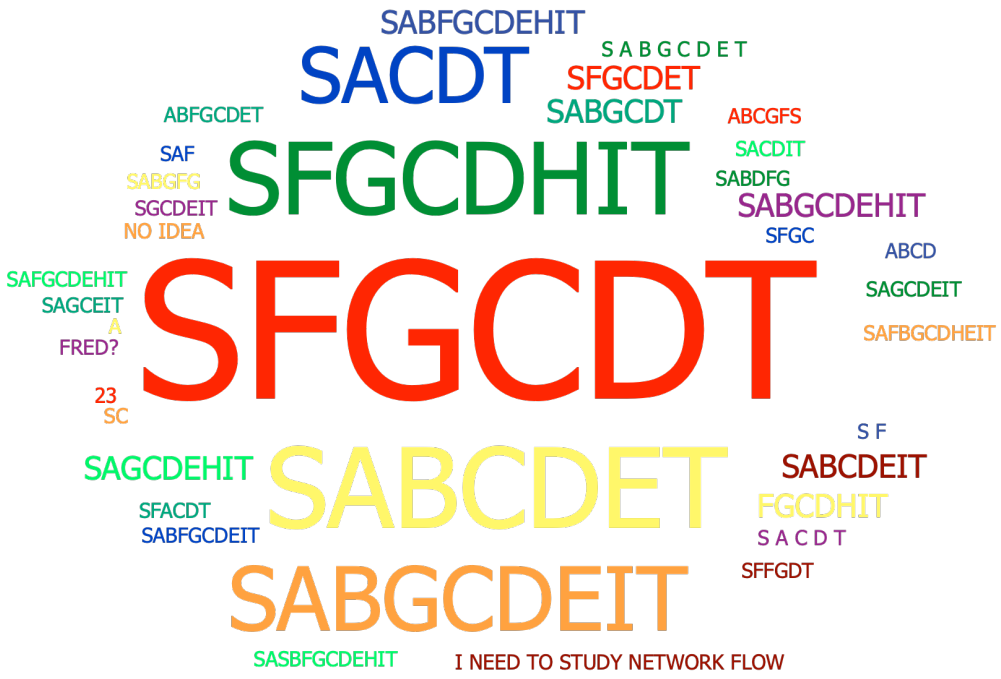
Responses

	Percent	Count
SFGCDT	18.6%	16
SABCDET	10.47%	9
SFGCDHIT	9.3%	8
SABGCDEIT	8.14%	7
SACDT	6.98%	6
FGCDHIT	2.33%	2
SABCDEIT	2.33%	2
SABFGCDEHIT	2.33%	2
SABGCDEHIT	2.33%	2
SABGCDT	2.33%	2
SAGCDEHIT	2.33%	2
SFGCDET	2.33%	2
23	1.16%	1
A	1.16%	1
ABCD	1.16%	1
ABCGFS	1.16%	1
ABFGCDET	1.16%	1
FRED?	1.16%	1
I NEED TO STUDY NETWORK FLOW	1.16%	1
NO IDEA	1.16%	1
S A B G C D E T	1.16%	1
S A C D T	1.16%	1
S F	1.16%	1
SABDFG	1.16%	1
SABFGCDEIT	1.16%	1
SABGFG	1.16%	1
SACDIT	1.16%	1
SAF	1.16%	1
SAFBGCDHEIT	1.16%	1
SAFGCDEHIT	1.16%	1

SAGCDEIT	1.16%	1
SAGCEIT	1.16%	1
SASBFGCDEHIT	1.16%	1
SC	1.16%	1
SFACDT	1.16%	1
SFFGDT	1.16%	1
SFGC	1.16%	1
SGCDEIT	1.16%	1
Totals	**100%**	**86**

Keyword(s): Any path that includes GC and has a flow of 23

Keyword Matches: 0



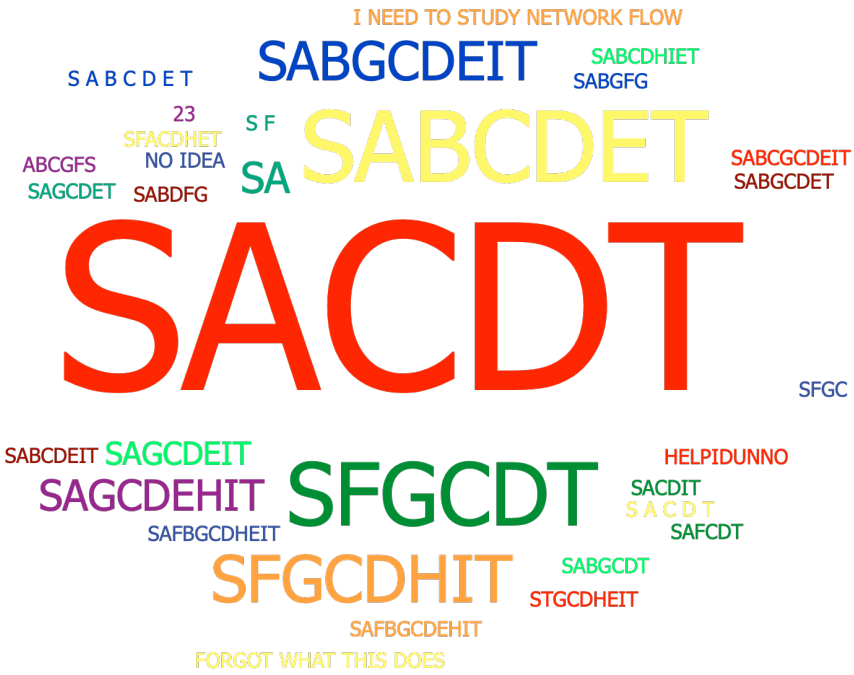
4. Please do question 4 which is on the screen in class. (Short Answer)

	Responses	
	Percent	**Count**
SACDT *(c)*	22.73%	20
SABCDET	9.09%	8
SFGCDT	7.95%	7
SFGCDHIT	5.68%	5
SABGCDEIT	4.55%	4
SA	3.41%	3
SAGCDEHIT	3.41%	3
SAGCDEIT	2.27%	2
23	1.14%	1
ABCGFS	1.14%	1
ACD	1.14%	1
ACDET	1.14%	1
F	1.14%	1
FGCDET	1.14%	1
FORGOT WHAT THIS DOES	1.14%	1
FRED!	1.14%	1
HELPIDUNNO	1.14%	1
I NEED TO STUDY NETWORK FLOW	1.14%	1
NO IDEA	1.14%	1
S A B C D E T	1.14%	1
S A C D T	1.14%	1
S F	1.14%	1
SABCDEIT	1.14%	1
SABCDET. I'M RICK HARRISON, AND THIS IS MY PAWN SHOP. I WORK HERE WITH MY OLD MAN AND MY	1.14%	1

SON, BIG HOSS. EVERYTHING IN HERE HAS A STORY AND A PRICE. ONE THING I'VE LEARNED AFTER 21 YEARS - YOU NEVER KNOW WHAT IS GONNA COME THROUGH THAT DOOR.		
SABCDHIET	1.14%	1
SABCDT	1.14%	1
SABCGCDEIT	1.14%	1
SABDFG	1.14%	1
SABGCDET	1.14%	1
SABGCDT	1.14%	1
SABGFG	1.14%	1
SACDIT	1.14%	1
SAFBGCDEHIT	1.14%	1
SAFBGCDHEIT	1.14%	1
SAFCDT	1.14%	1
SAGCDET	1.14%	1
SAGCDHIT	1.14%	1
SAGCDT	1.14%	1
SFACDHET	1.14%	1
SFGC	1.14%	1
SFGCDDT	1.14%	1
SFGCDET	1.14%	1
STGCDHEIT	1.14%	1
YABR	1.14%	1
Totals	**100%**	**88**

Keyword(s): SACDT

Keyword Matches: 20



5. Please do question 5 which is on the screen in class. (Short Answer)

	Responses	
	Percent	**Count**
CORRECT	18.6%	16
NO	12.79%	11
INCORRECT	8.14%	7
YES	6.98%	6
FALSE	3.49%	3
T	2.33%	2

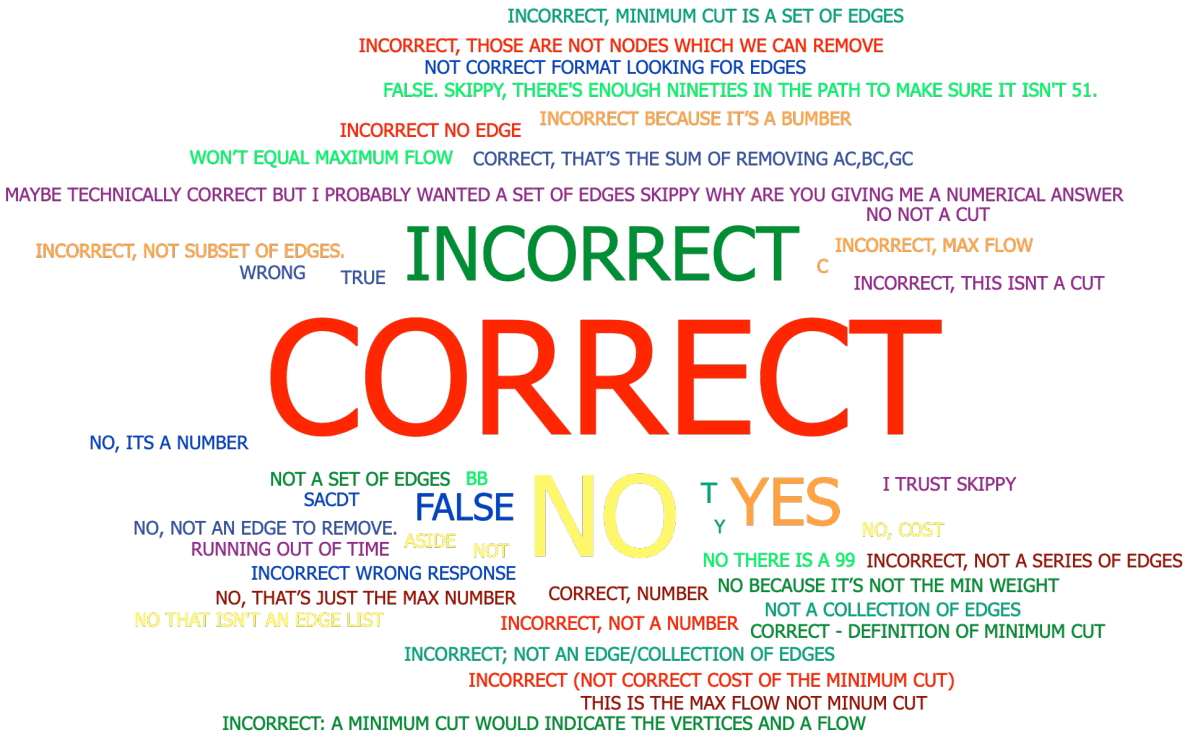
ASIDE	1.16%	1
BB	1.16%	1
C	1.16%	1
CORRECT - DEFINITION OF MINIMUM CUT	1.16%	1
CORRECT, NUMBER	1.16%	1
CORRECT, THAT'S THE SUM OF REMOVING AC,BC,GC	1.16%	1
FALSE. SKIPPY, THERE'S ENOUGH NINETIES IN THE PATH TO MAKE SURE IT ISN'T 51.	1.16%	1
I TRUST SKIPPY	1.16%	1
INCORRECT (NOT CORRECT COST OF THE MINIMUM CUT)	1.16%	1
INCORRECT BECAUSE IT'S A BUMBER	1.16%	1
INCORRECT NO EDGE	1.16%	1
INCORRECT WRONG RESPONSE	1.16%	1
INCORRECT, MAX FLOW	1.16%	1
INCORRECT, MINIMUM CUT IS A SET OF EDGES	1.16%	1
INCORRECT, NOT A NUMBER	1.16%	1
INCORRECT, NOT A SERIES OF EDGES	1.16%	1
INCORRECT, NOT SUBSET OF EDGES.	1.16%	1
INCORRECT, THIS ISNT A CUT	1.16%	1

INCORRECT, THOSE ARE NOT NODES WHICH WE CAN REMOVE	1.16%	1
INCORRECT: A MINIMUM CUT WOULD INDICATE THE VERTICES AND A FLOW	1.16%	1
INCORRECT; NOT AN EDGE/COLLECTION OF EDGES	1.16%	1
MAYBE TECHNICALLY CORRECT BUT I PROBABLY WANTED A SET OF EDGES SKIPPY WHY ARE YOU GIVING ME A NUMERICAL ANSWER	1.16%	1
NO BECAUSE IT'S NOT THE MIN WEIGHT	1.16%	1
NO NOT A CUT	1.16%	1
NO THAT ISN'T AN EDGE LIST	1.16%	1
NO THERE IS A 99	1.16%	1
NO, COST	1.16%	1
NO, ITS A NUMBER	1.16%	1
NO, NOT AN EDGE TO REMOVE.	1.16%	1
NO, THAT'S JUST THE MAX NUMBER	1.16%	1
NOT	1.16%	1
NOT A COLLECTION OF EDGES	1.16%	1
NOT A SET OF EDGES	1.16%	1
NOT CORRECT	1.16%	1

FORMAT LOOKING FOR EDGES		
RUNNING OUT OF TIME	1.16%	1
SACDT	1.16%	1
THIS IS THE MAX FLOW NOT MINUM CUT	1.16%	1
TRUE	1.16%	1
WON'T EQUAL MAXIMUM FLOW	1.16%	1
WRONG	1.16%	1
Y	1.16%	1
Totals	**100%**	**86**

Keyword(s): No - Number not cut.

Keyword Matches: 0



6. Please do question 6 which is on the screen in class. (Short Answer)

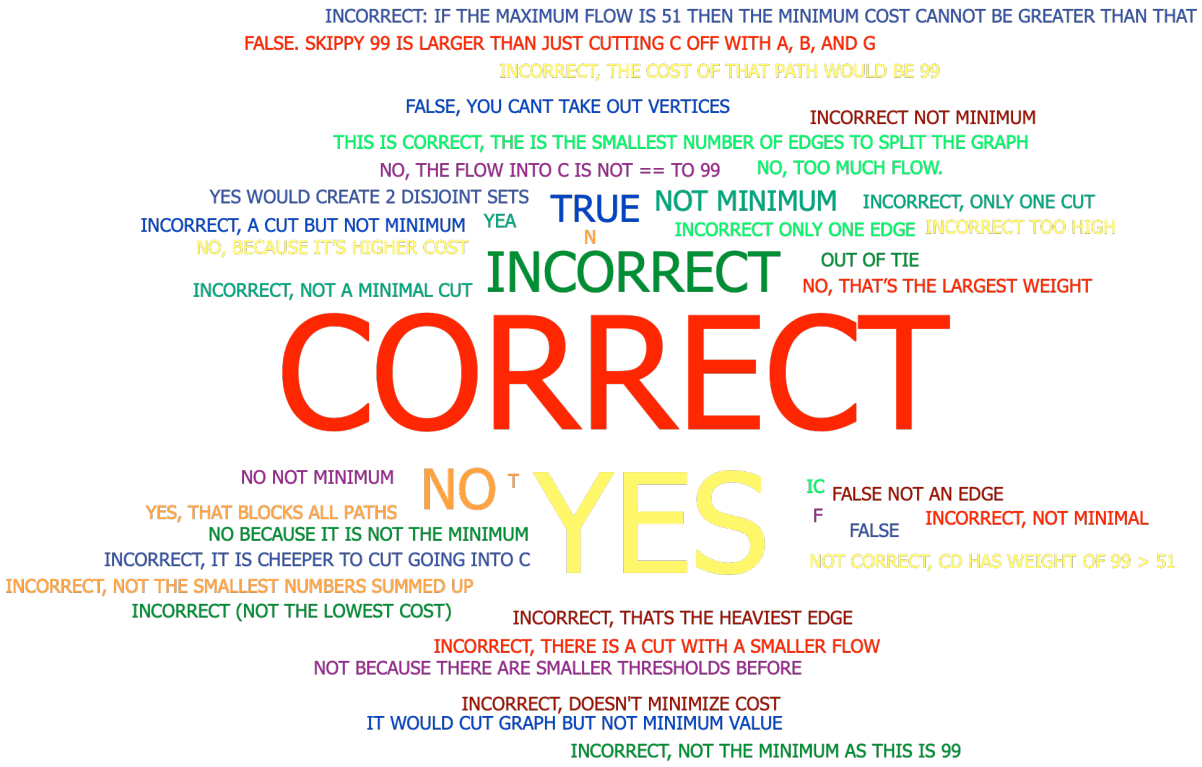
	Responses	
	Percent	**Count**
CORRECT	19.28%	16
YES	16.87%	14
INCORRECT	6.02%	5
NO	6.02%	5
TRUE	3.61%	3
NOT MINIMUM	2.41%	2
F	1.2%	1
FALSE	1.2%	1
FALSE NOT AN EDGE	1.2%	1
FALSE, YOU CANT TAKE OUT VERTICES	1.2%	1
FALSE. SKIPPY 99 IS LARGER THAN JUST CUTTING C OFF WITH A, B, AND G	1.2%	1
IC	1.2%	1
INCORRECT (NOT THE LOWEST COST)	1.2%	1
INCORRECT NOT MINIMUM	1.2%	1
INCORRECT ONLY ONE EDGE	1.2%	1
INCORRECT TOO HIGH	1.2%	1
INCORRECT, A CUT BUT NOT MINIMUM	1.2%	1
INCORRECT, DOESN'T	1.2%	1

MINIMIZE COST		
INCORRECT, IT IS CHEEPER TO CUT GOING INTO C	1.2%	1
INCORRECT, NOT A MINIMAL CUT	1.2%	1
INCORRECT, NOT MINIMAL	1.2%	1
INCORRECT, NOT THE MINIMUM AS THIS IS 99	1.2%	1
INCORRECT, NOT THE SMALLEST NUMBERS SUMMED UP	1.2%	1
INCORRECT, ONLY ONE CUT	1.2%	1
INCORRECT, THATS THE HEAVIEST EDGE	1.2%	1
INCORRECT, THE COST OF THAT PATH WOULD BE 99	1.2%	1
INCORRECT, THERE IS A CUT WITH A SMALLER FLOW	1.2%	1
INCORRECT: IF THE MAXIMUM FLOW IS 51 THEN THE MINIMUM COST CANNOT BE GREATER THAN THAT	1.2%	1
IT WOULD CUT GRAPH BUT NOT MINIMUM VALUE	1.2%	1
N	1.2%	1
NO BECAUSE IT IS NOT THE MINIMUM	1.2%	1
NO NOT MINIMUM	1.2%	1

NO, BECAUSE IT'S HIGHER COST	1.2%	1
NO, THAT'S THE LARGEST WEIGHT	1.2%	1
NO, THE FLOW INTO C IS NOT == TO 99	1.2%	1
NO, TOO MUCH FLOW.	1.2%	1
NOT BECAUSE THERE ARE SMALLER THRESHOLDS BEFORE	1.2%	1
NOT CORRECT, CD HAS WEIGHT OF 99 > 51	1.2%	1
OUT OF TIE	1.2%	1
T	1.2%	1
THIS IS CORRECT, THE IS THE SMALLEST NUMBER OF EDGES TO SPLIT THE GRAPH	1.2%	1
YEA	1.2%	1
YES WOULD CREATE 2 DISJOINT SETS	1.2%	1
YES, THAT BLOCKS ALL PATHS	1.2%	1
Totals	**100%**	**83**

Keyword(s): No - Cut but not minimum

Keyword Matches: 0



7. Please do question 7 which is on the screen in class. (Short Answer)

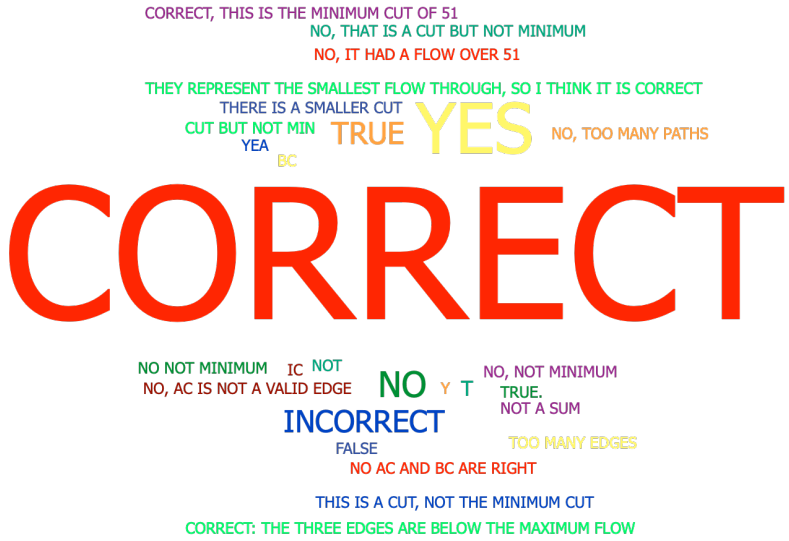
	Responses	
	Percent	**Count**
CORRECT	41.67%	35
YES *(c)*	14.29%	12
NO	5.95%	5
INCORRECT	4.76%	4
TRUE	4.76%	4
T	2.38%	2

BC	1.19%	1
CORRECT, THIS IS THE MINIMUM CUT OF 51	1.19%	1
CORRECT: THE THREE EDGES ARE BELOW THE MAXIMUM FLOW	1.19%	1
CUT BUT NOT MIN	1.19%	1
FALSE	1.19%	1
IC	1.19%	1
NO AC AND BC ARE RIGHT	1.19%	1
NO NOT MINIMUM	1.19%	1
NO, AC IS NOT A VALID EDGE	1.19%	1
NO, IT HAD A FLOW OVER 51	1.19%	1
NO, NOT MINIMUM	1.19%	1
NO, THAT IS A CUT BUT NOT MINIMUM	1.19%	1
NO, TOO MANY PATHS	1.19%	1
NOT	1.19%	1
NOT A SUM	1.19%	1
THERE IS A SMALLER CUT	1.19%	1
THEY REPRESENT THE SMALLEST FLOW THROUGH, SO I THINK IT IS CORRECT	1.19%	1
THIS IS A CUT, NOT THE MINIMUM CUT	1.19%	1
TOO MANY EDGES	1.19%	1

TRUE.	1.19%	1
Y	1.19%	1
YEA	1.19%	1
Totals	**100%**	**84**

Keyword(s): Yes

Keyword Matches: 12



8. Please do question 8 which is on the screen in class. (Short Answer)

	Responses	
	Percent	**Count**
NO	10.84%	9
YES	8.43%	7
INCORRECT	7.23%	6

CORRECT	6.02%	5
FALSE	3.61%	3
F	2.41%	2
NOT A CUT	2.41%	2
CORRECT: THE TWO PATHS ARE BELOW MAXIMUM FLOW	1.2%	1
DOESN'T CUT GRAPH(DT)	1.2%	1
DT STILL FLOWS	1.2%	1
DT STILL THERE	1.2%	1
EL	1.2%	1
FALSE, DOESN'T DISCONNECT	1.2%	1
FALSE. SKIPPY, THAT'S WORSE THAN JUST CUTTING CD. THAT'S 104 > 99 > 51	1.2%	1
IC	1.2%	1
INCORECT STILL GET TO T	1.2%	1
INCORRECT (CAN STILL GET FROM S TO T)	1.2%	1
INCORRECT - NOT MINIMUM	1.2%	1
INCORRECT DOES NOT DIVIDE THE GRAPH INTO TWO SETS	1.2%	1
INCORRECT DOESN'T DISCONNECT GRAPH	1.2%	1
INCORRECT STILL CONNECTED	1.2%	1
INCORRECT, D STILL GOES TO T	1.2%	1

INCORRECT, DOESNT COMPLETELY DISABLE GRAPH	1.2%	1
INCORRECT, DT	1.2%	1
INCORRECT, IT'S NOT A CUT	1.2%	1
INCORRECT, NOT A CUT	1.2%	1
INCORRECT, NOT A CUT.	1.2%	1
INCORRECT, PATH STILL EXISTS VIA DT AND IT	1.2%	1
INCORRECT, THERE IS STILL A PATH FROM S TO T	1.2%	1
INCORRECT, THERE IS STILL A ROUTE FROM S TO T	1.2%	1
INCORRECT, YOU CAN STILL REACH T FROM D	1.2%	1
INCORRECT. NOT SURE, HAVEN'T STUDIED MINIMUM CUT YET	1.2%	1
INCORRECT; T IS STILL CONNECTED TO D	1.2%	1
N	1.2%	1
NO 99 IS LESS	1.2%	1
NO BECAUSE DT ALSO EXISTS AND THERE ARE SMALLER EDGES BEFOREHAND	1.2%	1
NO BECAUSE IT DOES NOT CUT THE GRAPH IN TWO	1.2%	1

NO BECAUSE YOU ARE MISSING DT	1.2%	1
NO NOT A CUT	1.2%	1
NO NOT MIN CUT	1.2%	1
NO NOT MINIMUM	1.2%	1
NO THAT ISN'T A CUT	1.2%	1
NO, BECAUSE IT IS STILL CONNECTED	1.2%	1
NO, CAN'T BOTH BE MIN CUT	1.2%	1
NO, GRAPH STILL CONNECTED	1.2%	1
NO, NOT MINIMUM	1.2%	1
NO, STILL PATHS FOR POTENTIAL FLOW	1.2%	1
NOT	1.2%	1
NOT CORRECT, DOES NOT SEPARATE IN TWO COMPONENTS	1.2%	1
OR	1.2%	1
THERE ARE EDGES CONNECTED TO T STILL	1.2%	1
THERE IS A SMALLER CUT	1.2%	1
THERE IS STILL A PATH TO THE SINK	1.2%	1
THIS DOES NOTHING, S AND T ARE STILL CONNECTED	1.2%	1
TRUE	1.2%	1
WRONG. DT	1.2%	1

Keyword(s): No - Sum is 51, but not a cut

Totals	100%	83	Keyword Matches:	0

CORRECT

INCORRECT

F

NOT A CUT

NO

FALSE

YES

NO 99 IS LESS

FALSE, DOESN'T DISCONNEC

NO, STILL PATHS FOR POTENTIAL FLOW

CORRECT: THE TWO PATHS ARE BELOW MAXIMUM FLOW

NO BECAUSE IT DOES NOT CUT THE GRAPH IN TWO