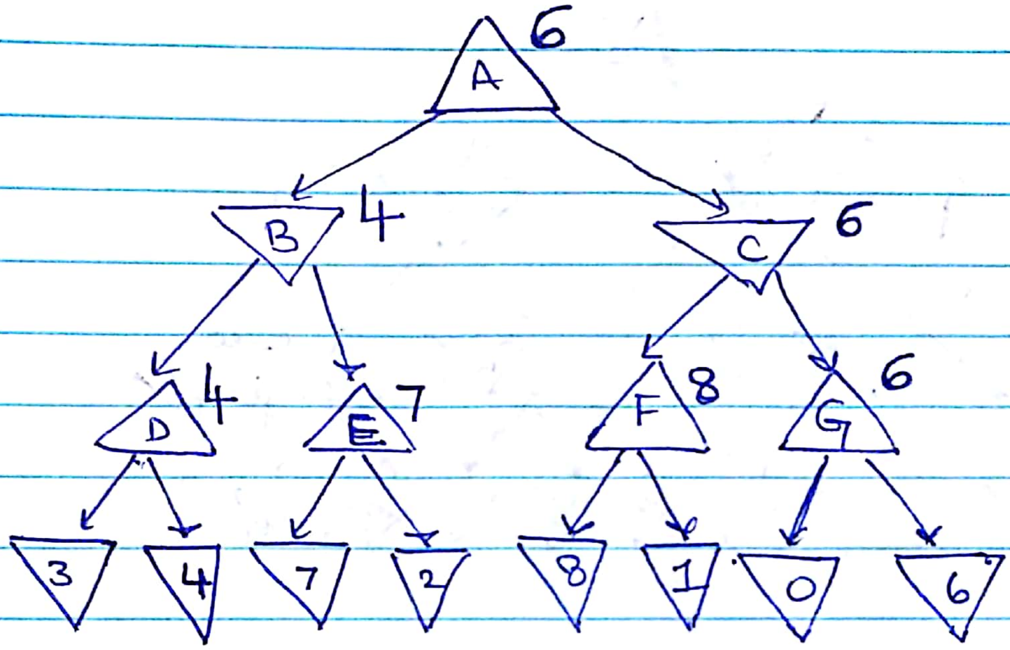
Utkarsh Jain
AI HW5

Question 1

1.



Game Theoretic Values

A : 6

B : 4

C : 6

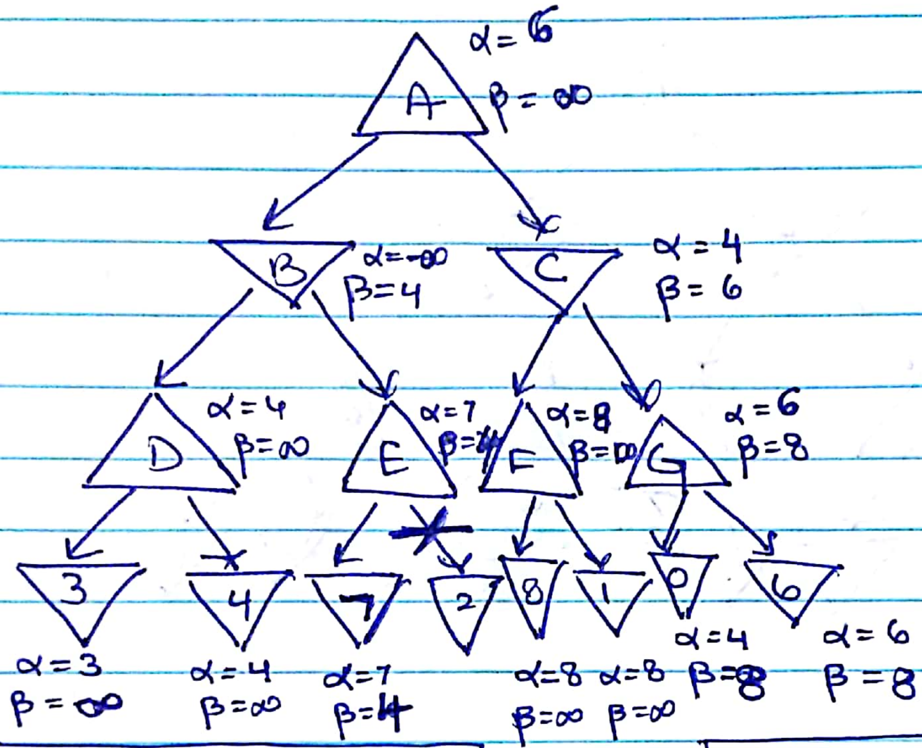
D : 4

E : 7

F : 8

G : 6

2.



A: $\alpha = 6$
$\beta = \infty$
Game Theoretic $= 6$

6: $\alpha = 6, \beta = 8$, G.T $= 6$

B: $\alpha = -\infty$
$\beta = 4$
Game Theoretic $= 4$

C: $\alpha = 4$
$\beta = 6$
Game Theoretic $= 6$

D: $\alpha = 4$
$\beta = \infty$
Game Theoretic $= 4$

E: $\alpha = 7$
$\beta = 4$
Game Theoretic $= 7$

0: $\alpha = 4, \beta = 8$, G.T $= 0$

F: $\alpha = 8$
$\beta = \infty$
Game Theoretic $= 8$

1: $\alpha = 8, \beta = \infty$, G.T $= 1$

G: $\alpha = 6$
$\beta = 8$
Game Theoretic $= 6$

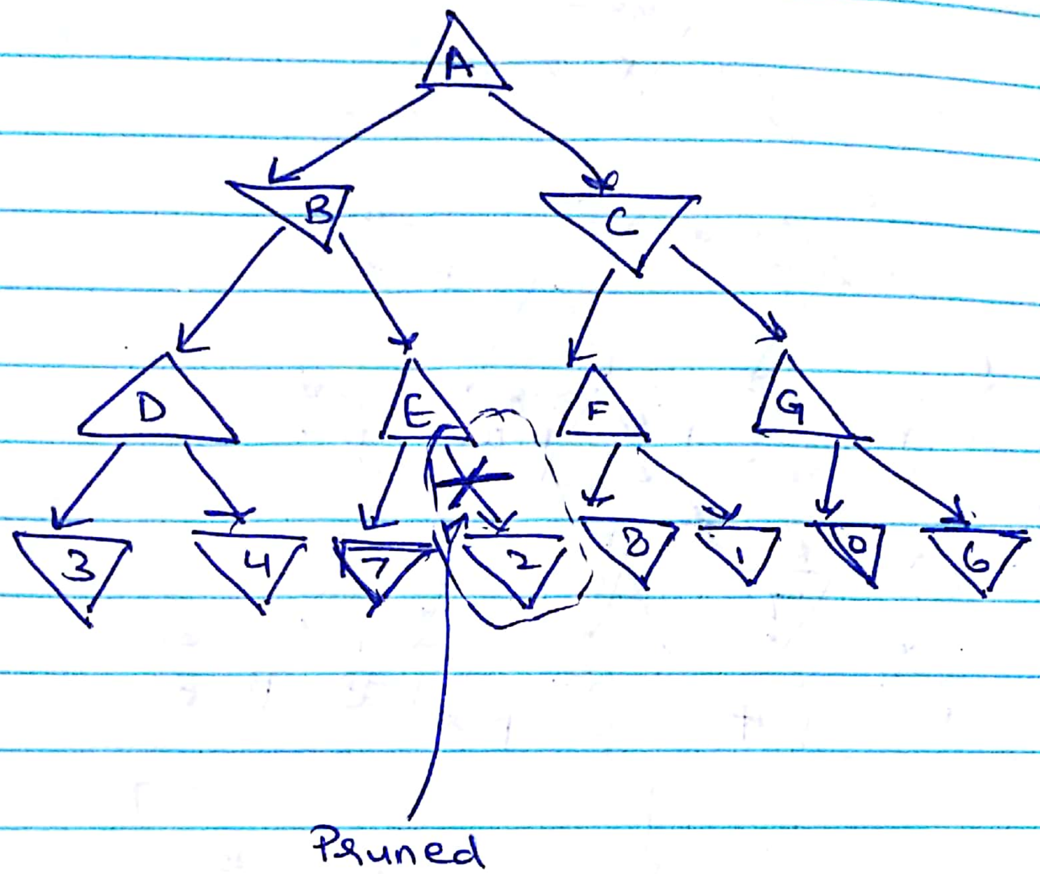
8: $\alpha = 8, \beta = \infty$, G.T $= 8$

3: $\alpha = 3, \beta = \infty$, G.T $= 3$

4: $\alpha = 4, \beta = \infty$, G.T $= 4$

7: $\alpha = 7, \beta = 4$, G.T $= 7$

3.



The branch that connects E to 2 will be pruned as when we are at E after visiting node 7, the α value becomes 7 and $\beta = 4$ and thus $\alpha \geq \beta$. This means the max player would never allow the game to move towards this path so it is not necessary to inspect the other children of E.

Scanned by CamScanner

4. By using alpha-beta pruning we decrease the amount of time taken by not having to inspect reduntant parts of the tree. If $\alpha \geq \beta$, then the game would never take that path and thus the rest of the children of the node where $\alpha \geq \beta$ will not provide a more optimal path. Thus we reduce our search space by pruning so we dont have visit all nodes and time taken by our program decreases. Even though alpha beta pruning will not always result in not having to search the whole tree, there will be a lot of cases where branches will get pruned.

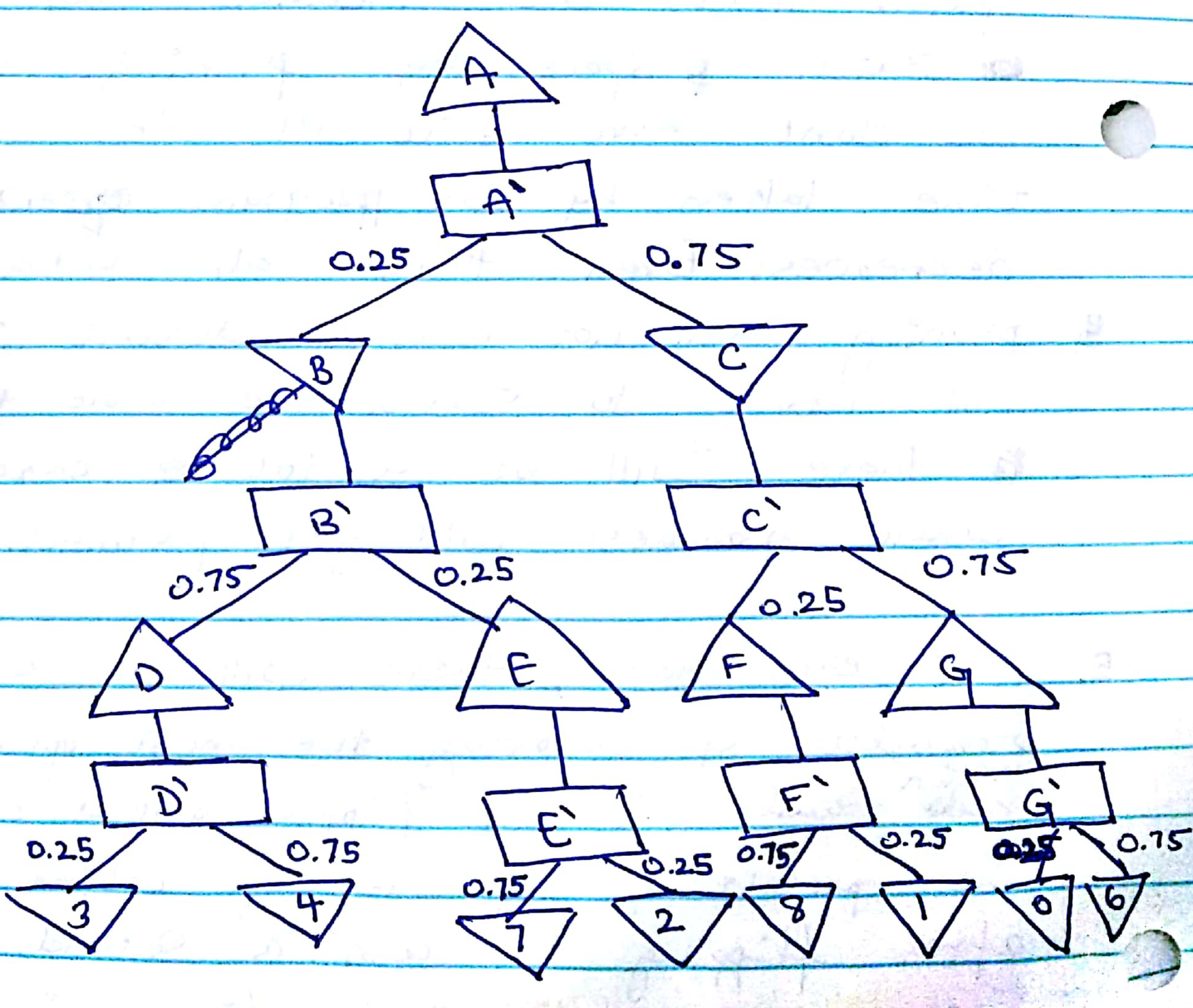
5. At each node, there will be 0.5 probability of choosing the most optimal action and 0.5 of flipping another coin. The probability of getting optimal action after flipping the coin a second time Since we got tails in the first coin flip will be p(tails in first) * p(optimal action chosen in second)

$= 0.5 \times 0.5 = 0.25$.

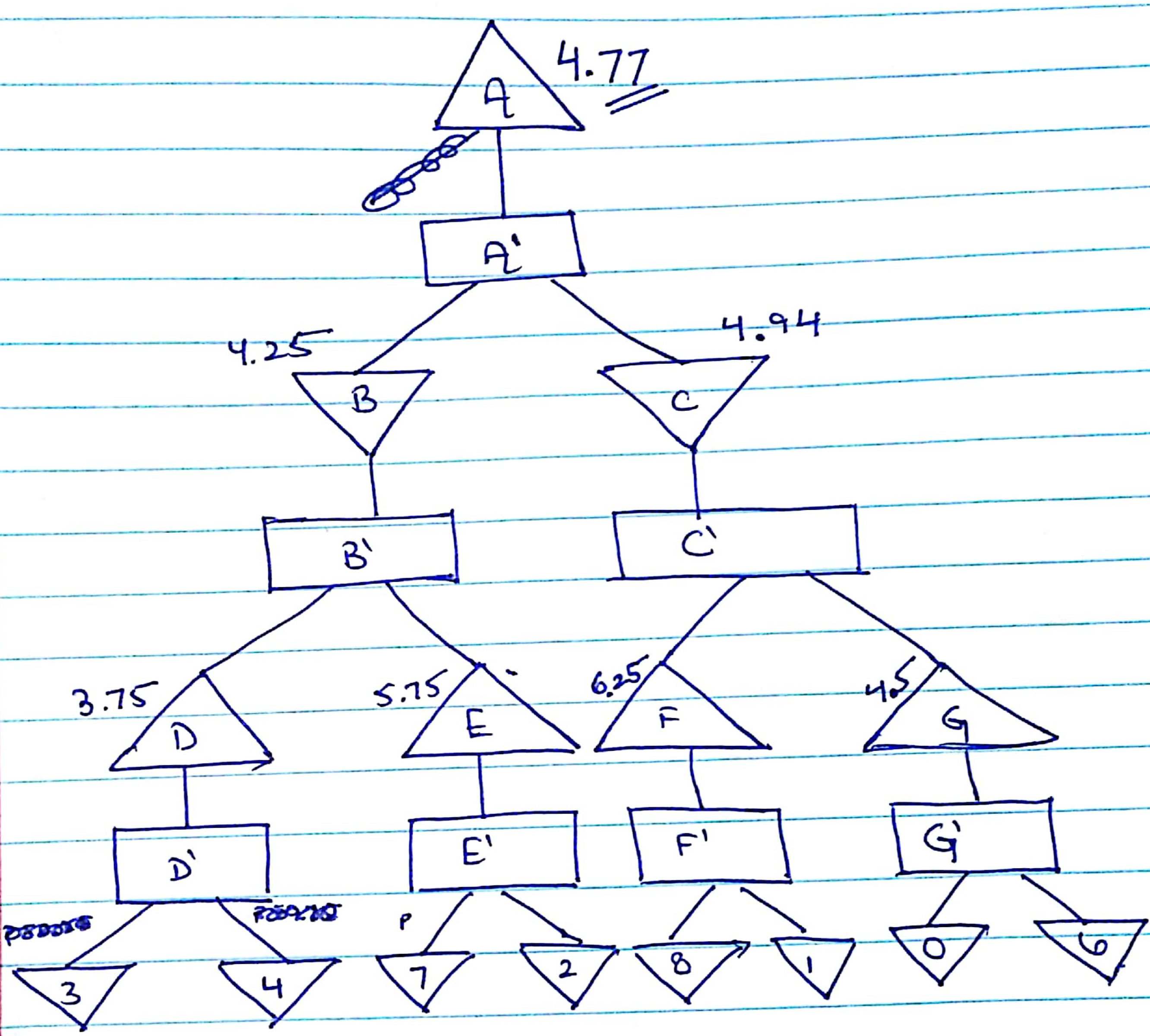
So total probability of choosing the most optimal action is $0.5 + 0.25 = \underline{\underline{0.75}}$

At each node probability of choosing an optimal action is $\boxed{0.75.}$

6.



7.



Expected Game Theoretic Values

D: $0.25 \times 3 + 0.75 \times 4 = 3.75$

E: $0.75 \times 7 + 0.25 \times 2 = 5.75$

F: $0.75 \times 8 + 0.25 \times 1 = 6.25$

G: $0.25 \times 0 + 0.75 \times 6 = 4.5$

B: $3.75 \times 0.75 + 0.25 \times 5.75 = 4.25$

C: $0.25 \times 6.25 + 4.5 \times 0.75 = 4.94$

A: $0.25 \times 4.25 + 0.75 \times 4.94 = 4.77$

AI
HW 5

Question 2

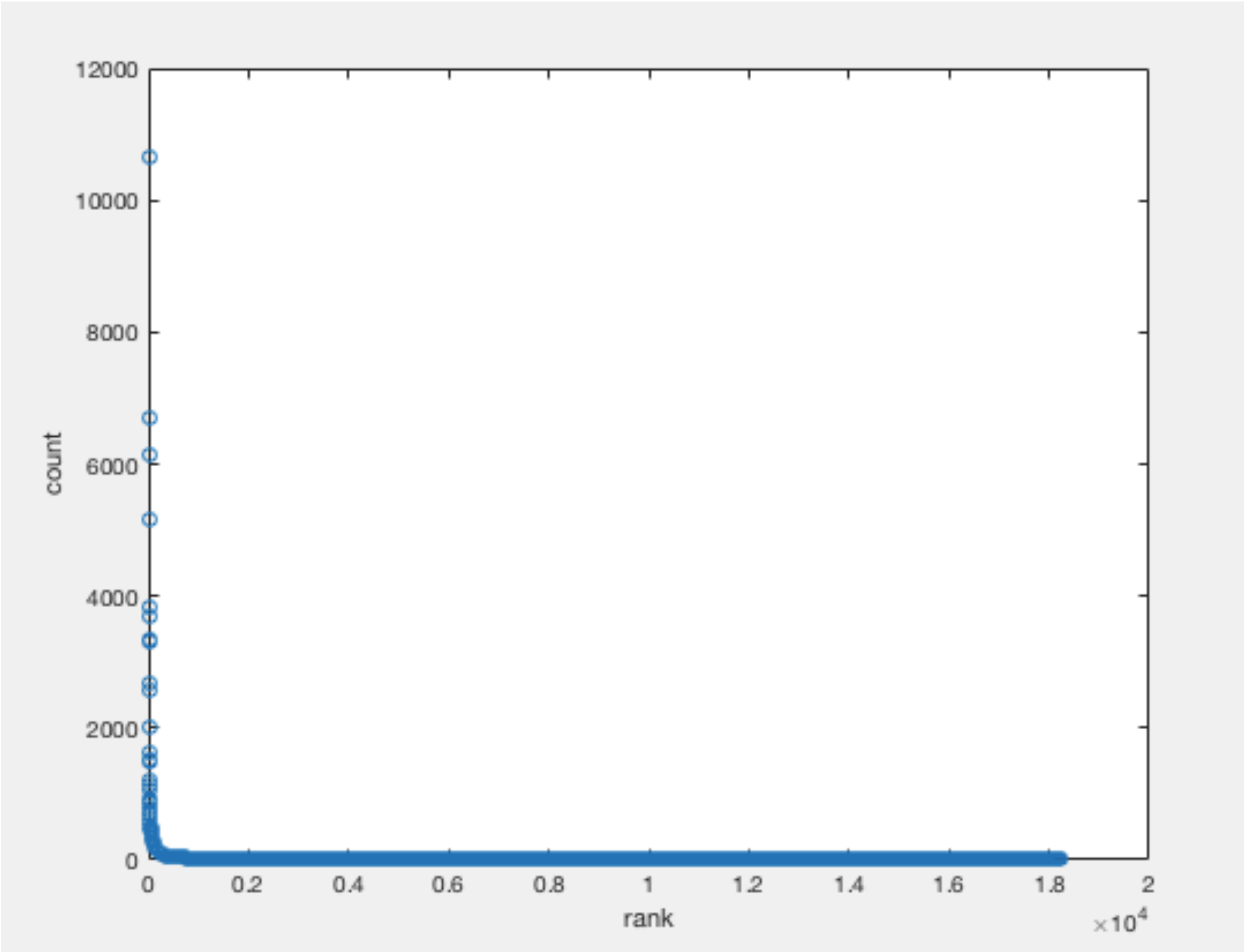
1. The most convenient way that I can think of for a computer program to break the corpus into words is using whitespaces as a delimiter. Basically assume that words are separated by whitespaces to count the words in the corpus. I used the Scanner in Java.

2. There are 205117 word tokens and 18263 word types in the corpus.

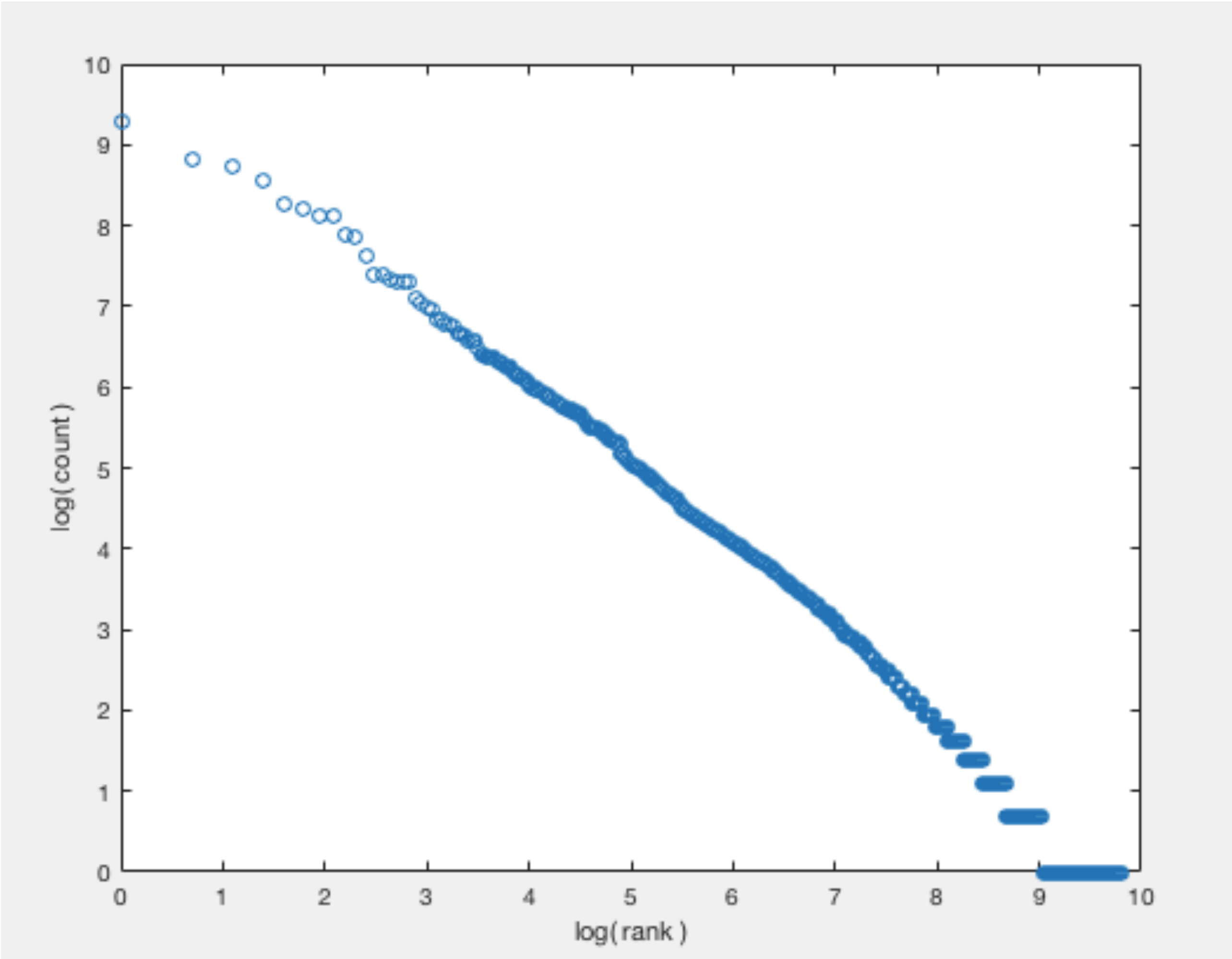
3.
Count: 10636 Word: the
Count: 6686 Word: to
Count: 6128 Word: of
Count: 5157 Word: and
Count: 3846 Word: in
Count: 3702 Word: a
Count: 3354 Word: that
Count: 3312 Word: is
Count: 2663 Word: be
Count: 2567 Word: AI
Count: 2022 Word: will
Count: 1615 Word: for
Count: 1613 Word: are
Count: 1517 Word: it
Count: 1505 Word: not
Count: 1483 Word: as
Count: 1483 Word: on
Count: 1221 Word: with
Count: 1152 Word: The
Count: 1066 Word: have

4.
Count: 1 Word:“leading
Count: 1 Word:“law
Count: 1 Word:“it”
Count: 1 Word:“interactive,
Count: 1 Word:“intelligent.”
Count: 1 Word:“intelligence”,
Count: 1 Word:“intelligence”
Count: 1 Word:“information
Count: 1 Word:“increased
Count: 1 Word:“in
Count: 1 Word:“imminent
Count: 1 Word:“images
Count: 1 Word:“human-aware
Count: 1 Word:“historical
Count: 1 Word:“growth
Count: 1 Word:“gradually
Count: 1 Word:“goal-directed
Count: 1 Word:“getting
Count: 1 Word:“get
Count: 1 Word:“firstcarquote”

5.


12000
10000
8000
6000
4000
2000
0
count
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
rank
×10^4

6.



7.

The shape of curve shown in part 5 shows that there are very few words which have extremely high counts. Most words have a low count. The counts of the words decrease very rapidly.

The shape of the curve shown in part 6 is close to a straight line. The fact that the log of the x axis and y axis gives a straight line shows that the the relation r*c = e^(x) [where x is some constant] is approximately true.

8.

Issue 1:
Using my definition of computer words, uppercase and lowercase letters are treated differently even though they have the same meaning. Words such as “The” and “the” should be treated as the same word for most Natural Language Processing applications.

Issue 2:
[“intelligence”,], [“intelligence”] and [“intelligent.”] are treated as different words even though they all have the same root and same meaning. [“intelligence”,] and [“intelligence”] are the same exact word but are being treated as different words due to the punctuation that is coming after them. The punctuation should ideally not be part of the token and either be treated as a separate token or not counted at all. This is another major issue with my definition of the computer word.