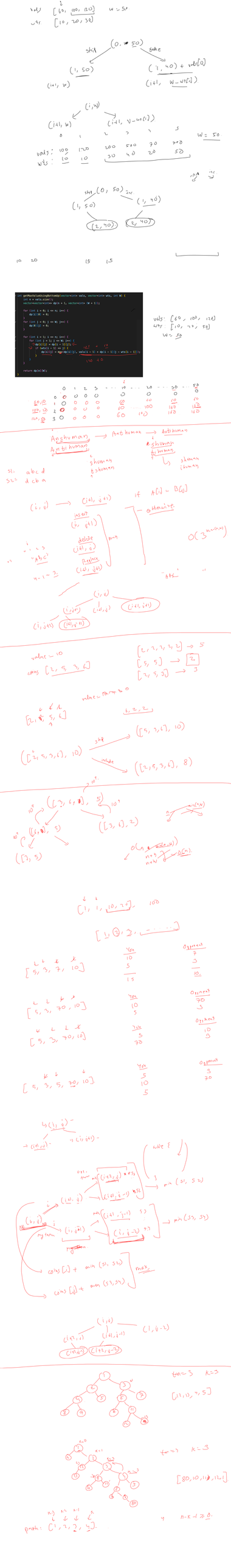
int getMaxValueKnapSack(vector<int> vals, vector<int> wts, int W) {
int n = vals.size();
vector<vector<int>> dp(n + 1, vector<int> (W + 1));
for (int i = 0; i <= n; i++) {
dp[i][0] = 0;
}
for (int j = 0; j <= W; j++) {
dp[0][j] = 0;
}
for (int i = 1; i <= n; i++) {
for (int j = 1; j <= W; j++) {
// dp[i][j] = dp[i - 1][j];
if (wts[i - 1] <= j) {
dp[i][j] = max(dp[i][j], vals[i - 1] + dp[i - 1][j - wts[i - 1]]);
}
}
}
return dp[n][W];
}
vals: [60, 100, 120]
wts: [10, 20, 30]
W = 50
0 1 2 3 10 20 30 50
0 0 0 0 0 0 0 0
60,10 1 0 0 0 60 60 60
100,20 2 0 0 0 60 100 160 160
120,30 3 0 0 0 60 100 120 160