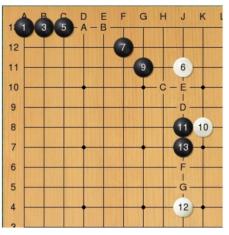
## Theory of Computer Games Homework # 6

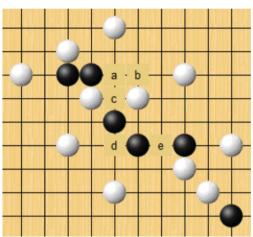
311605004

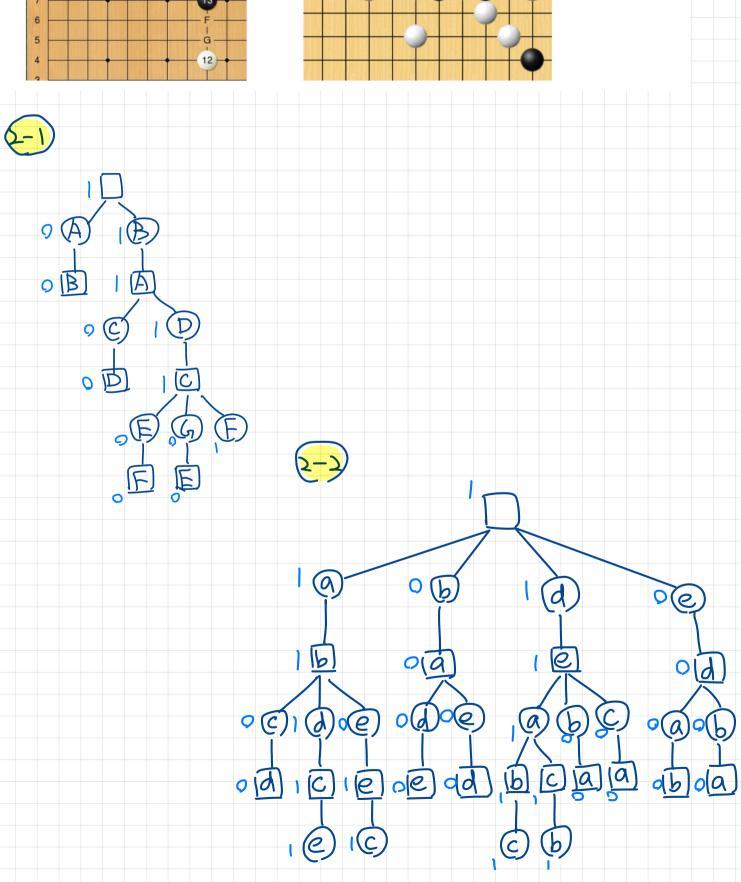
\_\_\_\_\_\_

1. For DLP game with "aabcdeabcdbcdeabcdee", draw the dependency-based search tree to solve this. Also describe how many nodes to explore, if use the traditional search with and without transposition table respectively. aabcdeabcdbcdeabcdee aabcdeabcdbcdeabcd bbcdeabcdbcdeabcdee ccdeabcdbcdeabcdee ddeabcdbcdeabcdee ecabcdbcdeabcdee aabcdeabcdbcdeaa aabcdbcdeabcdee aabcdeabcdbcdee aabcdeabcdbcdd ccdbcdeabcdee aabcdeabcdbcc dd bcdeabcdee aabcdeabcdbb Cbcdeabcdee aabcdeabcd c 4 using transition table: 10×10 = 1001 nodes (Passing each 10 possibilities on Jeft & right) If not using transition table: 2°-11 nodes depth =20

2. Consider the Gomoku game. For the lower left case, where Black to move, depict its  $\lambda^1$ -tree and also evaluate its value. For the lower right case, where Black to move and to win, depict its  $\lambda^2$ -tree.







3. If black is played at m10, depict all the next  $\lambda^2$ -moves by white.

a>>b>>c>>d>>e>>f>>g>>h>>i j>>k> l>>m>>n>>o

