Dijstras:  
 Standard heap, O(mlog(n))

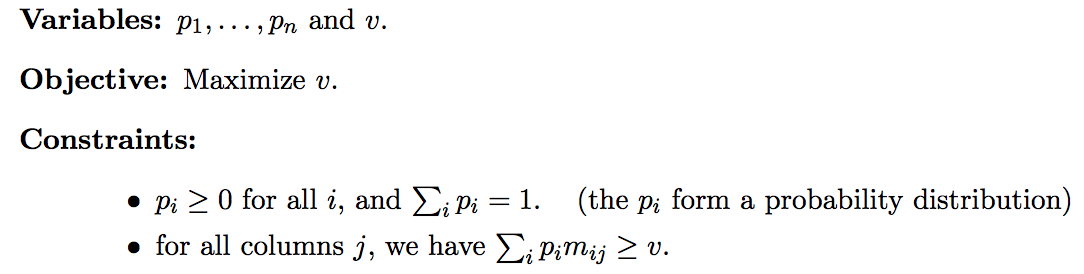
Fibonachi heap – O(m+ nlog(n))

All Shortest paths:

* O(nm log n) – potentials w/ dijstras – beats O(N^3) assuming sufficiently sparse
* Matrix Products – n^3log(n)
* Floyd Warshall – increasing # of intermediate nodes – O(n^3)-- for k = 1 to n do: for each i,j do: A[i][j] = min( A[i][j], (A[i][k] + A[k][j]);

To model flow in LP:

* Need capacity constraints
* Need flow conservation (flow in= flow out)
* Min-cost-max-flow – 2 step process or pump process

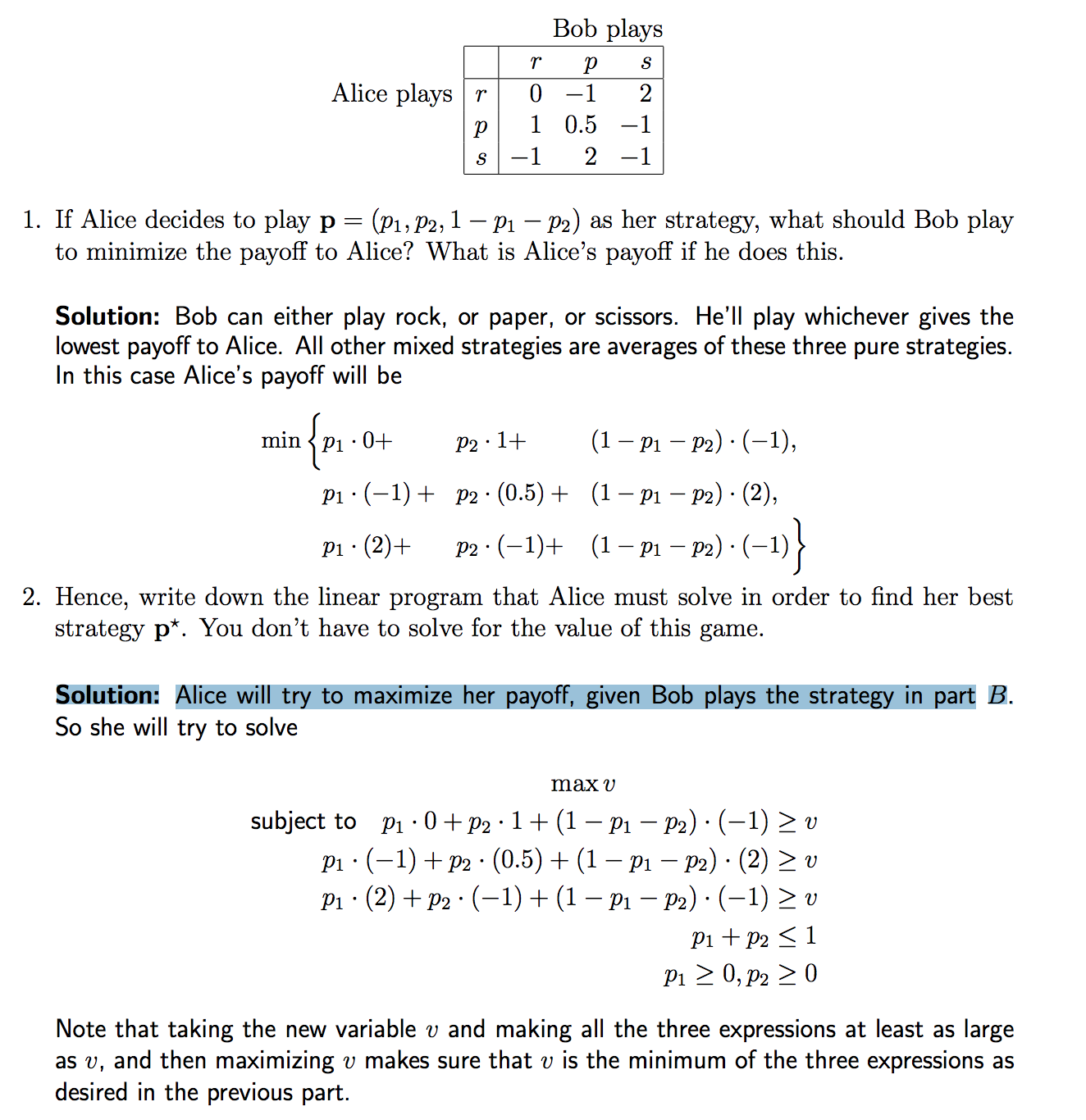
LPs to solve Games: 

LP Algos:  
 Simplex: Repeatedly look at neighbor and go to the best one

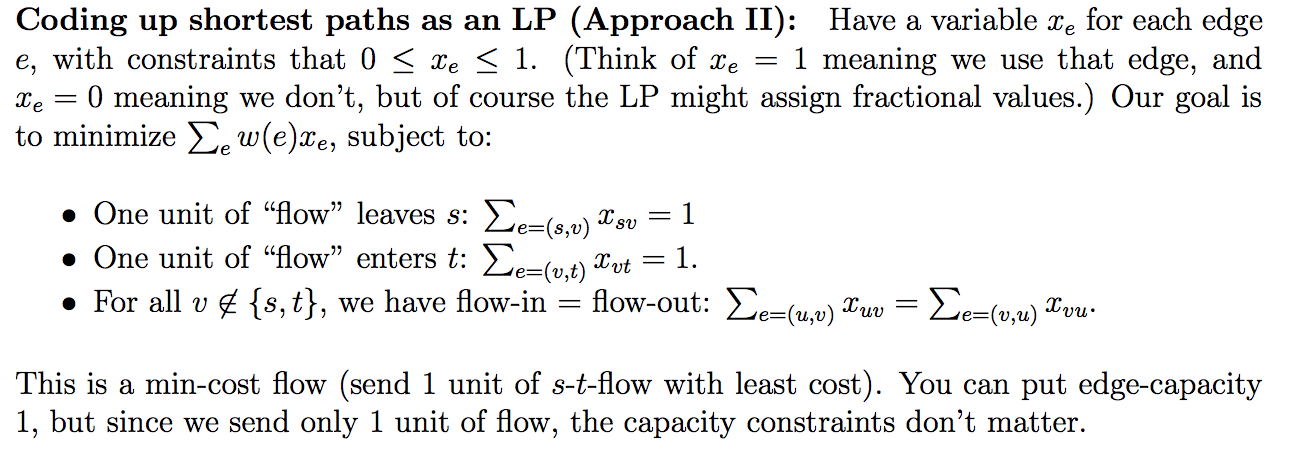
Elipsoid: Create an elipse and until a constraint is violated, find smaller elipse ---One can show that in each step, you can always create a new smaller ellipse whose volume is smaller, by at least a (1 − 1/n) factor, than the original ellipse

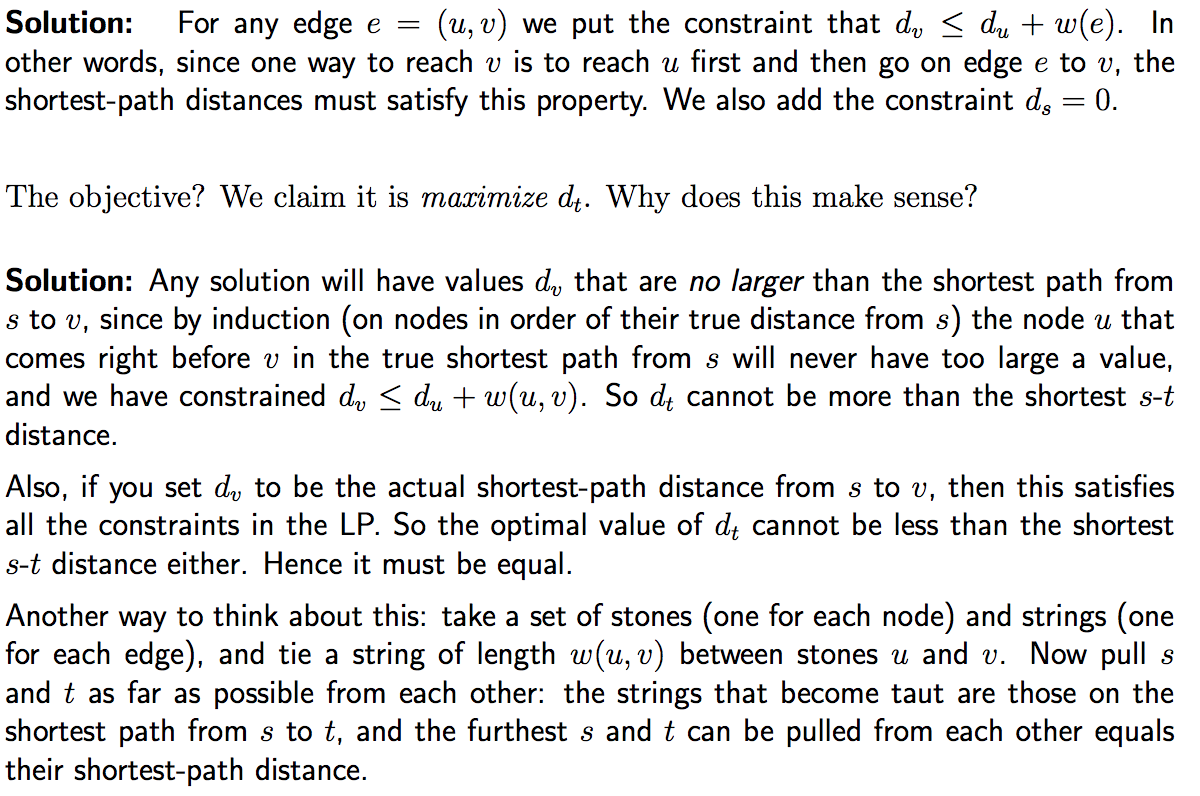
LP:

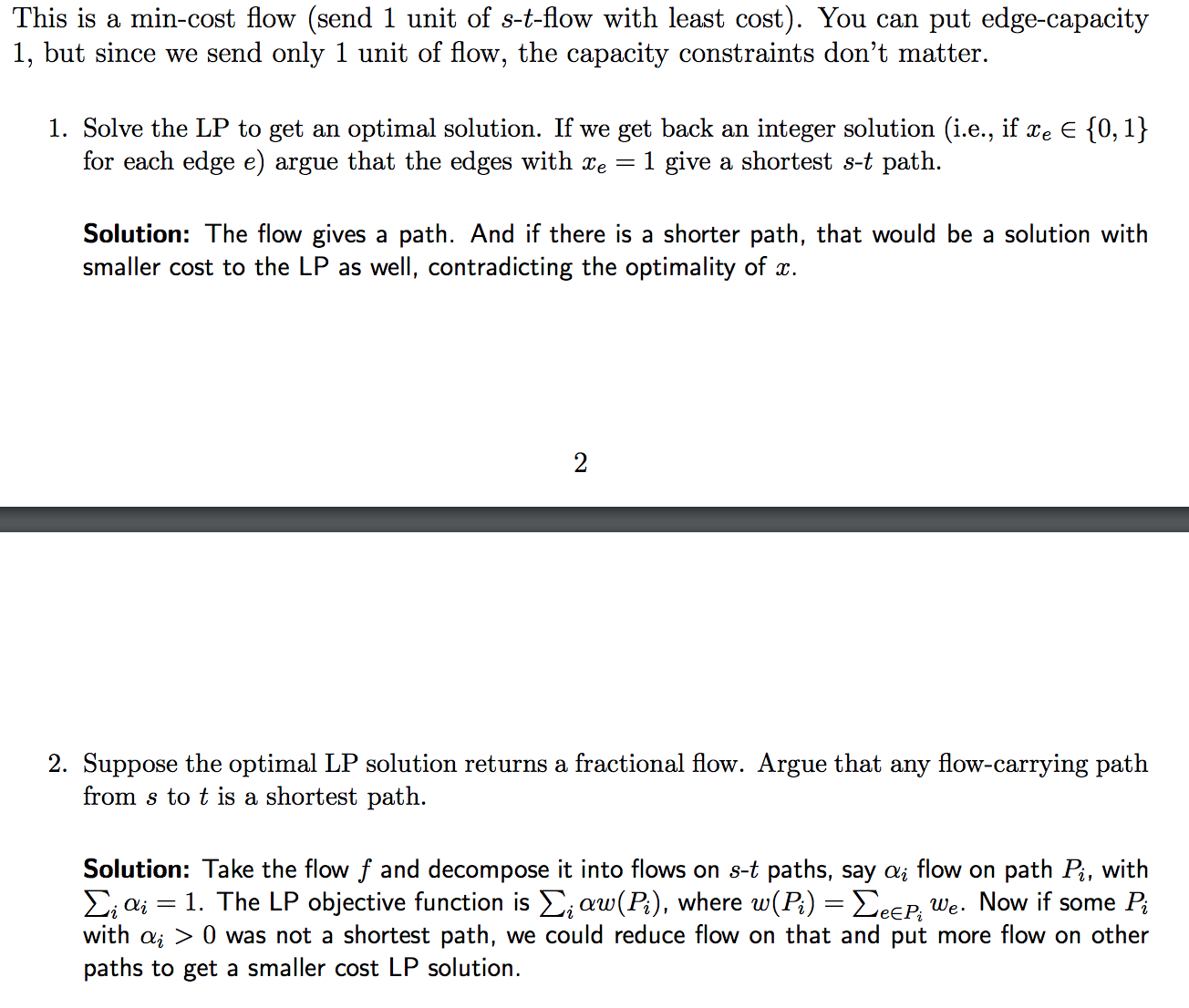
* Maximize your payoff when designing an LP
* Minimize your opponents payoff when deriving the strategy to min



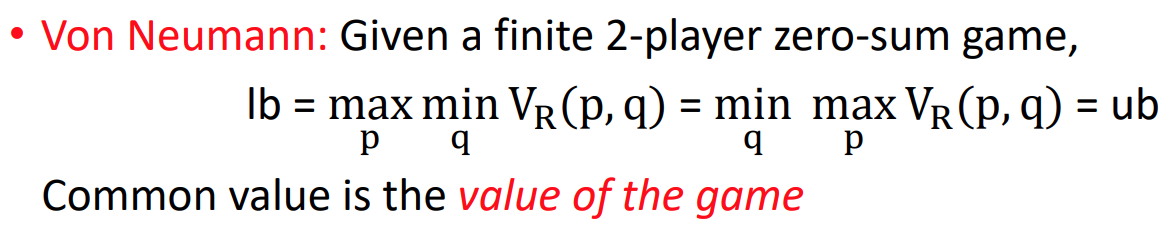
Shortest path LPs:





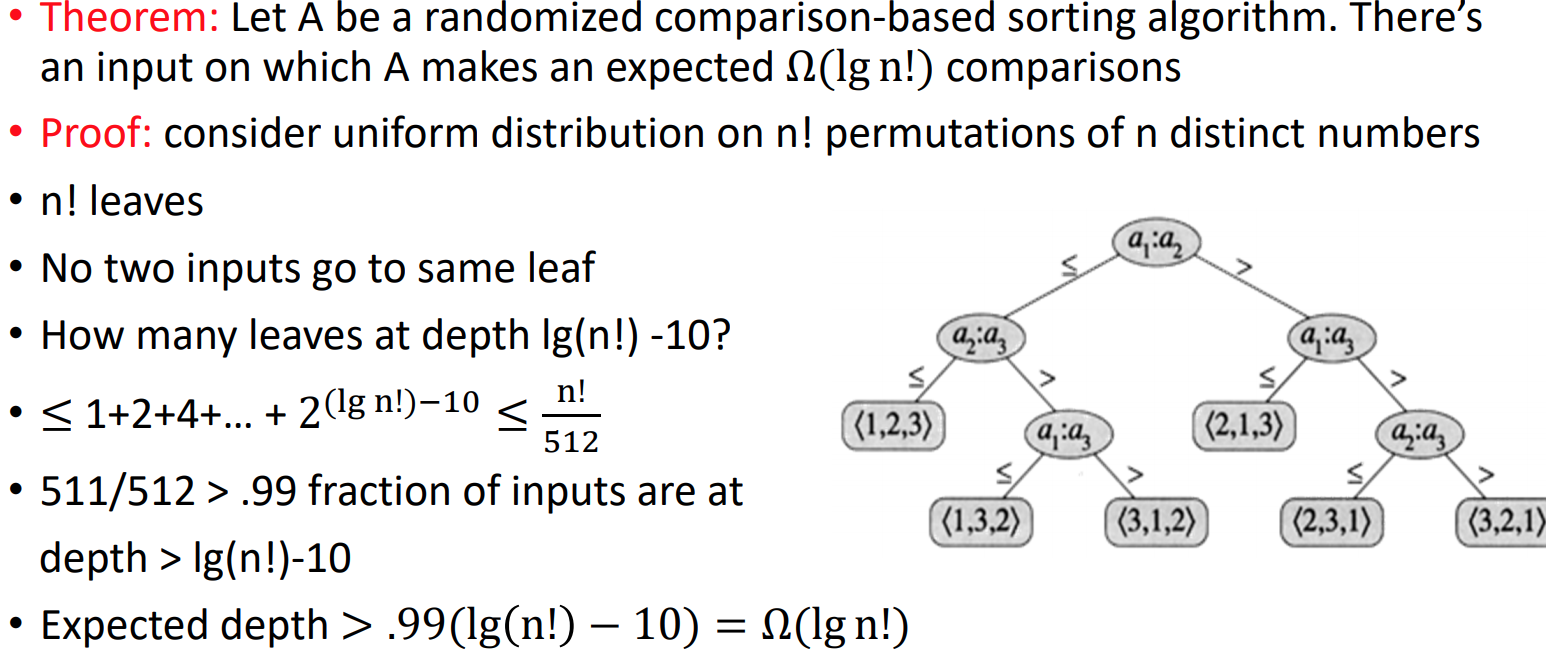


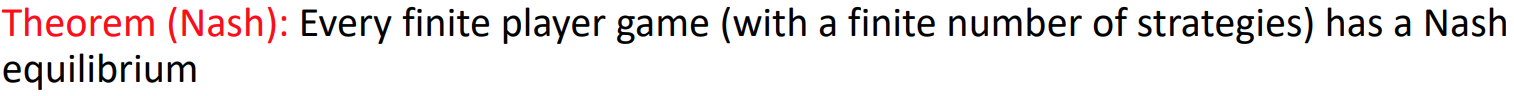
Reduce From – trying to solve the problem you are reducing FROM using a call to the problem you are reducing TO



LOWER BOUNDS RANDOM Algorithms (as games): find a distribution on columns (possible algorithms) such that the expected cost of each row is small

Example:





Subset-DP

Don’t understand last 2 pages of lecture 18

-----------------------------------------------------------------------------

Polytime:

