Maximum Flow w/o costs

What to study (in order)			**Problems**				
Ford Fulkerson theorem			Level 1	Level 2	Level 3	Level 4	Assorted
Edmonds algorithm			http://www.spoj.pl/problems/POTHOLE/	http://www.spoj.pl/problems/MUDDY/	http://www.spoj.pl/problems/BABY/	http://www.spoj.pl/problems/DIVREL/	PKU-3498
All the applications in here: http://en.wikipedia.org/wiki/Maximum_flow_problem			http://www.spoj.pl/problems/QUEST4/	http://www.spoj.pl/problems/ANGELS/	http://www.spoj.pl/problems/SCITIES/	http://codeforces.com/contest/132/problem/E	UVA-259
Menger Theorem			http://www.spoj.pl/problems/MTOTALF/	http://www.spoj.pl/problems/FASTFLOW/	http://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1675		UVA-10594
Konig Theorem (also, demonstrate konig theorem by minimum cut and see the maximum independent set problem)			http://www.spoj.pl/problems/DISJPATH/	http://www.spoj.pl/problems/COCONUTS/	http://codeforces.com/problemset/problem/78/E		PKU-1273
Min cost flow			http://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1452	http://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=2865	SRM 358 div 1000		UVA-10735
Dilworth theorem			SRM 303 div1 500	http://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1321	http://www.spoj.pl/problems/OILCOMP/		UVA-11506
Circulations (very high level)				http://mirrorarchive.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=365&page=show_problem&problem=2480	Google Code Jam 2008 – Beta - Hexagon		UVA-563
				Google Code Jam 2008 – Round 3 – Problem C	http://uva.onlinejudge.org/index.php?option=onlinejudge&Itemid=99999999&category=287&page=show_problem&problem=3244		UVA-10746
Reference implementations				http://poj.org/problem?id=2060	http://poj.org/problem?id=1637		PKU-1459
Dinic algorithm	http://pastebin.com/QjDQ4nV4						UVA-11082
Edmonds	http://pastebin.com/T1DND9Y0						UVA-11082
Min cost flow	http://pastebin.com/ECZAi3Rp						UVA-753
							UVA-10806
							PKU-2125
Reference ideas							UVA-11380
Everytime the problem needs to minimize something -> think about min cut, not max flow							http://acm.tju.edu.cn/toj/showp1119.html
Everytime the problem needs to maximize something -> think about max flow, not min cut							UVA-820
If the problem is about distributing something, think about flows							UVA-11301
If the problem says about partitioning something into two groups, yeah, draw a bipartite graph and run min cut							UVA-11506
If the problem is on a grid, yeah, a grid is a bipartite graph, so try something in the bipartite graph							UVA-11765
If the problem wants to minimize paths in a DAG, flow to the rescue! it can even have min cost :)							PKU-1087
In some problems, the edges can become vertices of a bipartite graph E x V							UVA-10380
A grid can be modeled as a bipartite graph in two ways:							UVA-11381
	1. A left vertex for each row and a right vertex for each column						PKU-1149
	2. As a grid can be colored like a chessboard, white and black cells are vertices of a bipartite graph						UVA-10249
Non directed graphs can be modeled for flows just by adding two directed edges with same capacity for each non-directed edge							
If vertices have capacities, each vertex has to be splitted into two vertices with an inner edge with the desired capacity. In addition, one vertex will be used for in-edges and the other for out-edges when communicating with other vertices							
In min cost max flow, if the initial graph contains no negative cycles, the augmenting procedure, when modifying the graph, will never create a negative cycle!							
In min cost max flow, if the initial graph contains negative cycles, brace yourself and run, run fast away from the contest							
Usually, if the problem doens't seem to be DP, try to draw a graph and run max flow							
Memorize all the implementations, please							

Books		
http://www.mediafire.com/?04mjyxde7mo	Jungnickel, 3rd edition	Suggested
	Algorithm Design, Eva Tardos	Preferred