TTK23 - Assignment 1 – Simen Keiland Fondevik

Problem 1: Value Iteration

a), b)

Gamma = 1.0 gives the following value function and policy.

0.812	0.868	0.918	1.000	\rightarrow	\rightarrow	\rightarrow	
0.762		0.660	-1.000	т		Т	
0.705	0.655	0.611	0.388	т	←	←	←

c)

See figures. With greater discount factor (bigger gamma) it is easier to see the +2 terminal state even when close to the +1 state. Thus, gamma = 0.99 finds a better policy.

Gamma = 0.9

										_										
0.154	0.172	0.192	0.211	0.189	0.169	0.151	0.135	0.121	0.105		←	1	←	+	1	1	1	1	1	←
0.172	0.192	0.216	0.239	0.212	0.185	0.166	0.145	0.131	0116		Ţ	1	1	1	1	1	+	+	1	←
0.192	0.216	0.242	0.272								→	→	1	1						
0.212	0.241	0.272	0.307	0.349	0.392	0.440	0.495	0.556	0.624		→	1								
0.189			0.273						0.705		1			1						1
0.169	0.190	0.214	0.242		0.775	0.876	0.984		0.792		→	→	→	1		→	→	1		1
0.155	0.173	0.193	0.214		0.690		1111		0.890		→	→	→	1		1		↓		1
0.170	0.190	0.211	0.193	-2.000	0.491		1249		1000		↓	↓	↓	1		1		↓		
0.190	0.213	0.240	0.269	0.305	0.431		1403				→	→	→	→	1	1		Ţ		
0.205	0.235	0.265	0.299	0.335	0.381		1576	1.780	2000		→	→	→	→	→	1		→	→	
										4										

Gamma = 0.99

1.323	1314	1324	1.335	1321	1307	1293	1280	1266	1.252	1	1	1	1	+	1	1	1	+	+
1.339	1.327	1.339	1.352	1336	1.320	1.305	1290	1275	1.261	+	→	+	→	Ţ	1	1	1	ļ	1
1.355	1341	1353	1365							1	1	1	1						
1371	1.355	1365	1.385	1369	1.354	1.339	1324	1310	1.294	+	1	→	+	1	1	1	1	1	1
1.387			1.402						1.250	1			1						1
1.403	1.416	1428	1415		1.826	1847	1565		1.266	1	1	1	1		→	→	1		1
1.415	1.432	1.445	1.433		1.506		1890		1.252	1	1	1	1		1		Ţ		1
1.433	1445	1463	1.445	-2.000	1585		1912		1.000	1	1	1	1		1		↓		
1445	1464	1481	1495	1516	1565		1933			†	→	→	→	1	1		↓		
1461	1479	1496	1514	1531	1550		1955	1978	2.000	→	→	→	→	→	1		→	→	

d)

With gamma = 1.0 we should from any state converge towards the +2 reward. Thus all states should have a reward close to 2 and the policy becomes kind of random in the cases where neighboring states are equally good. This is confirmed by the following results. As shown, the policy is unable to reach any terminal state.

1982	1962	1963	1963	1963	1983	1963	1983	1963	1963	1		Ţ	1	1	1	1	1	Ţ	1	Ţ
1963	1963	1984	1984	1984	1984	1984	1984	1984	1984	1		1	↓	↓	1	1	1	1	1	+
1984	1964	1964	1983							1		1	↓	1						
1985	1964	1985	1986	1986	1986	1986	1986	1986	1986	1		→	†	↓	Ţ	ļ	1	1	ļ	↓
1985			1987						1.965	1				→						1
1986	1987	1987	1965		2.000	2.000	2000		1985	1		+	+	→		→	1	1		1
1987	1965	1985	1988		2000		2000		1985	1		+	1	1		1		1		1
1965	1965	1989	1989	2,000	1992		2000		1000	1		1	1	+		→		1		
1965	1989	1989	1990	1991	1992		2000			[-	,	→	→	→	1	1		1		
1989	1989	1990	1990	1991	1992		2000	2000	2.000		,	→	→	→	→	1		1	1	

Problem 2: Policy Iteration

a)

I choose Iterative Policy Evaluation because it was the first algorithm in the assignment. Well, it also looks easier :P

b)

gamma = 1.0. Identical with 1a).

0.812	0.868	0.918	1.000	→	→	→	
0.762		0.660	-1.000	Т		Т	
0.705	0.655	0.611	0.388	Т	←	←	+

c)

Policy iteration can change policy to an equally good one for each iteration, thus never satisfy the termination requirement of two consecutive identical policies.

d)

Value iteration uses 13 iterations and its biggest error (except from changing terminal states from 0 to $\pm 1/-1$) was 0.792 on the tiny grid. For policy iteration the corresponding numbers are 7 iterations and 8.156 error.

Problem 3

Adding a negative reward of -0.01 for every non-terminal state finds a useful and optimal policy. As can be seen below, we are guaranteed to end in the +2 reward even when next to the +1 reward.

_										
	1501	1493	1501	1510	1499	1.489	1475	1467	1.457	1446
	1512	1504	1512	1522	1510	1499	1.487	1475	1464	1452
	1524	1514	1523	1535						
	1536	1524	1535	1546	1535	1524	1513	1502	1491	1479
	1548			1559						1465
	1559	1569	1577	1570		1909	1921	1932		1457
	1570	1580	1589	1581		1895		1944		1446
	1581	1591	1601	1.592	-2.000	1684		1955		1000
	1591	1602	1614	1625	1638	1671		1966		
	1601	1612	1624	1636	1645	1659		1977	1989	2,000

-										
	1	1	1	1		1	↓	↓	←	ţ
	+	+	+	+	1	1	1	1	1	1
	1	1	1	1						
	1	1	→	1	1	1	1	1	1	1
	1			1						1
	1	1	1	1		→	→	1		1
	1	1	1	1		1		+		1
	1	1	1	+		→		1		
	→	→	→	→	↓	1		↓		
	→	→	→	→	→	1		→	→	
-										