

Bruno Baffini Santana - 11834050  
Ilan Matheus Silva de Sousa - 11832920

Configurações da máquina utilizada nos testes:

Macbook Pro (13-inch, Mid 2012)

SO macOS Mojave 10.14.6

Processador 2,5 Ghz Intel Core i5

Memória 8Gb 1600Mhz DDR3

Memória secundária SSD SATA 240Gb

O código base para o grafo foi baseado no código desenvolvido nas aulas do professor Norton pela Univesp, o código da fila de prioridade foi adaptado de um EP desenvolvido durante a disciplina AED I, os algoritmos foram desenvolvidos a partir das aulas de AED II postadas na playlist da disciplina no youtube.

Bellman Ford							
n/p	0,10	0,20	0,40	0,50	0,60	0,80	0,90
100	0,000579	0,001244	0,002626	0,002686	0,005092	0,005306	0,005114
1.000	0,774480	1,675781	2,872141	2,651894	5,166007	5,280157	5,099313
1.500	3,017772	4,822778	9,775247	9,596904	16,937708	16,888153	16,875491
2.000	6,468795	9,865166	20,297088	20,425210	39,323665	39,658240	39,632996
3.000	18,857532	31,940386	72,148804	72,092885	142,058313	140,940752	140,791735
5.000	71,812789	137,152505	327,710881	324,335905	611,947966	608,638525	609,428174
10.000	525,194788	1.024,078119	2.447,725685	2.458,818936	-	-	-

Dijkstra							
n/p	0,10	0,20	0,40	0,50	0,60	0,80	0,90
100	0,000135	0,000181	0,000319	0,000305	0,000278	0,000315	0,000267
1.000	0,028488	0,033964	0,039512	0,039933	0,046572	0,047752	0,048618
1.500	0,090897	0,105080	0,127033	0,119531	0,132978	0,131574	0,129507
2.000	0,171592	0,207669	0,245787	0,253271	0,270643	0,258674	0,263868
3.000	0,470389	0,607008	0,708776	0,630702	0,693977	0,680556	0,764177
5.000	1,566867	1,965378	2,165572	2,033900	2,165588	2,123993	2,147574
10.000	10,200529	10,396018	11,522531	13,266956	-	-	-

Bellman Ford em DAGs

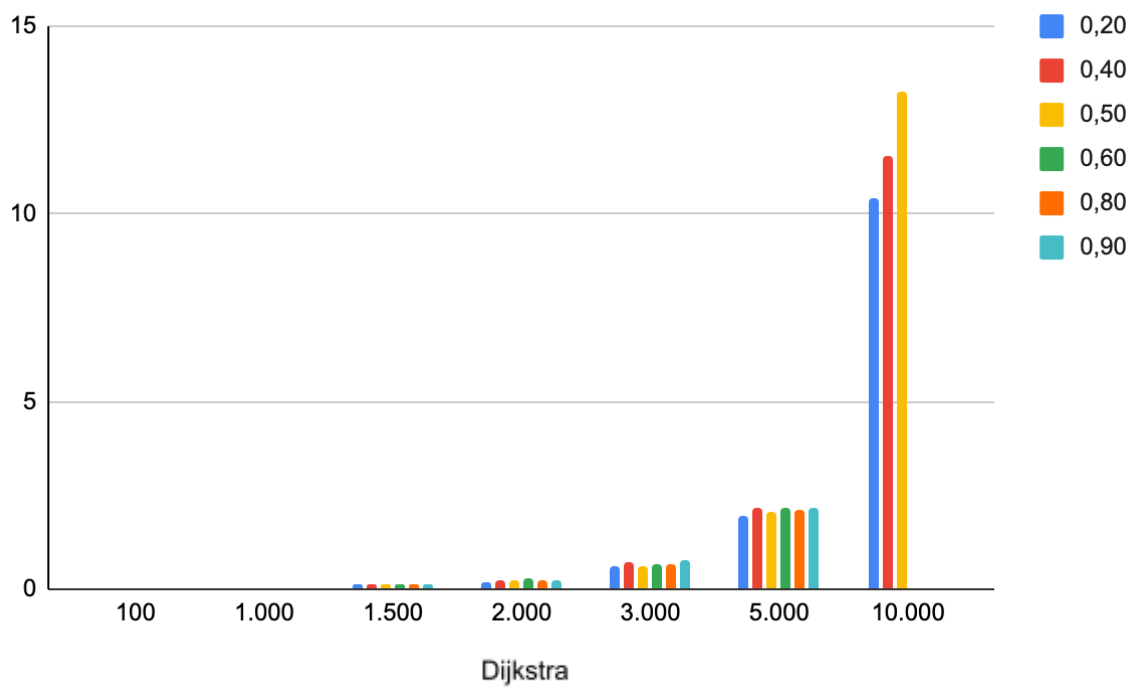
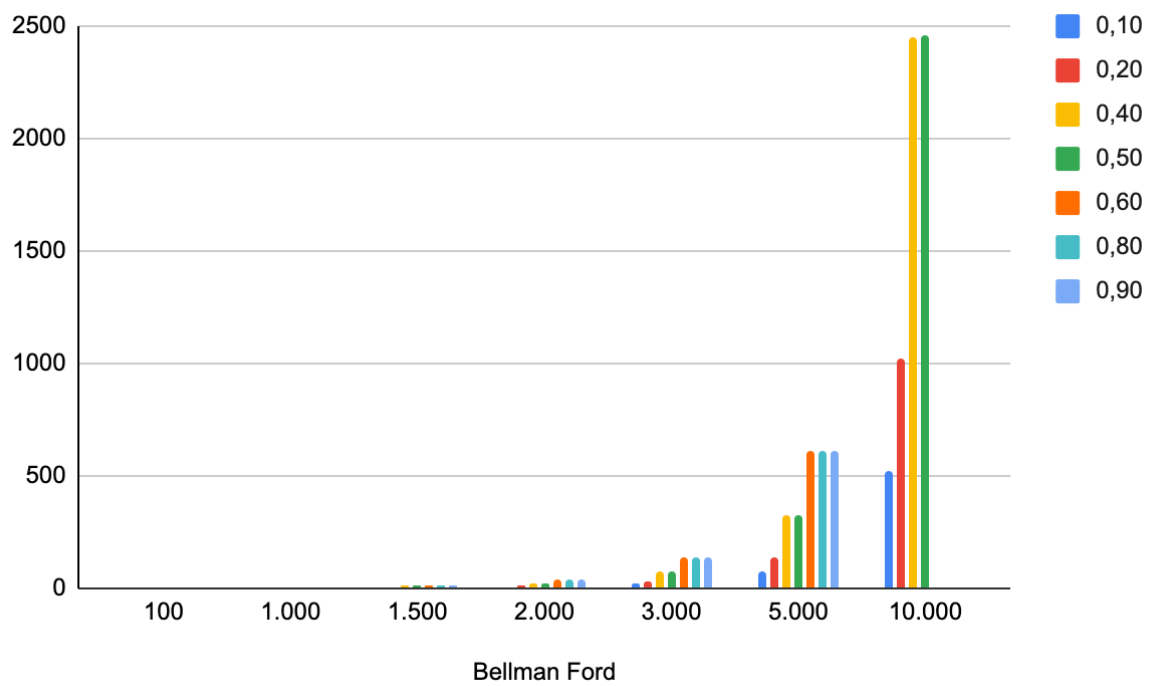
n\p	0,10	0,20	0,40	0,50	0,60	0,80	0,90
100	0,000288	0,000657	0,001465	0,001276	0,002561	0,002440	0,002694
1.000	0,285754	0,773596	1,816724	1,739627	2,906642	2,938893	2,910405
1.500	1,414601	2,729085	5,129534	5,246975	9,265775	9,226199	9,297563
2.000	3,618124	6,030173	11,661716	11,704000	21,220694	21,141543	21,129809
3.000	11,059554	17,159956	36,733699	37,057712	72,747963	74,471408	73,322654
5.000	43,532285	76,597073	171,569615	170,444953	331,626535	327,393912	323,411766
10.000	313,471002	573,790723	-	-	-	-	-

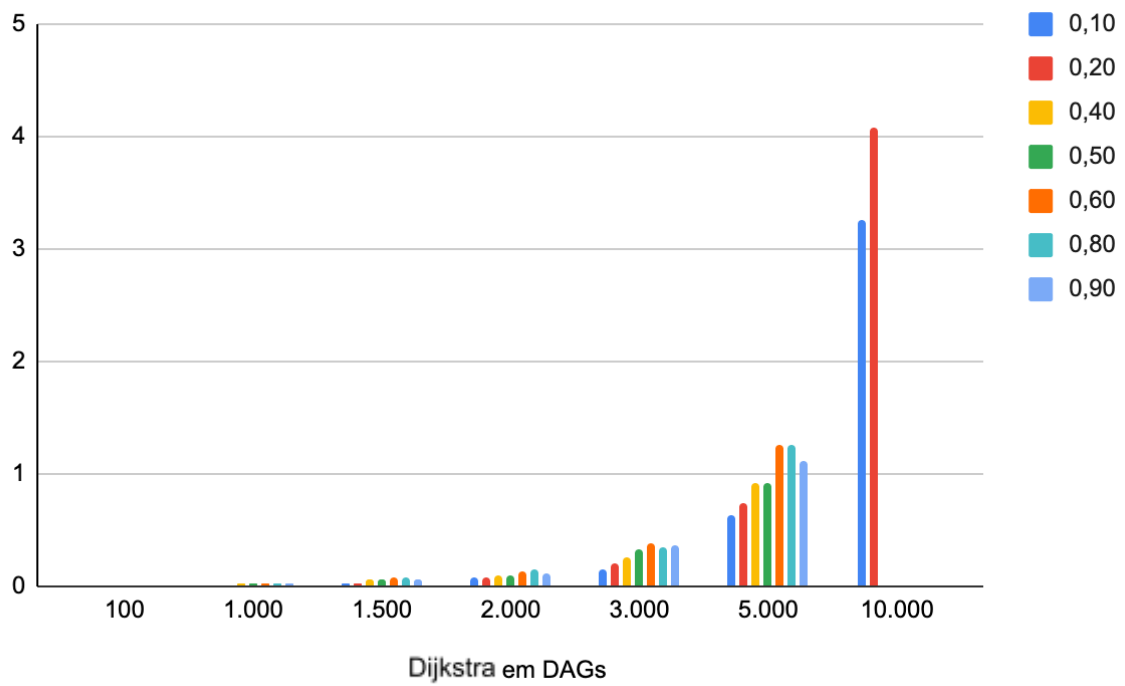
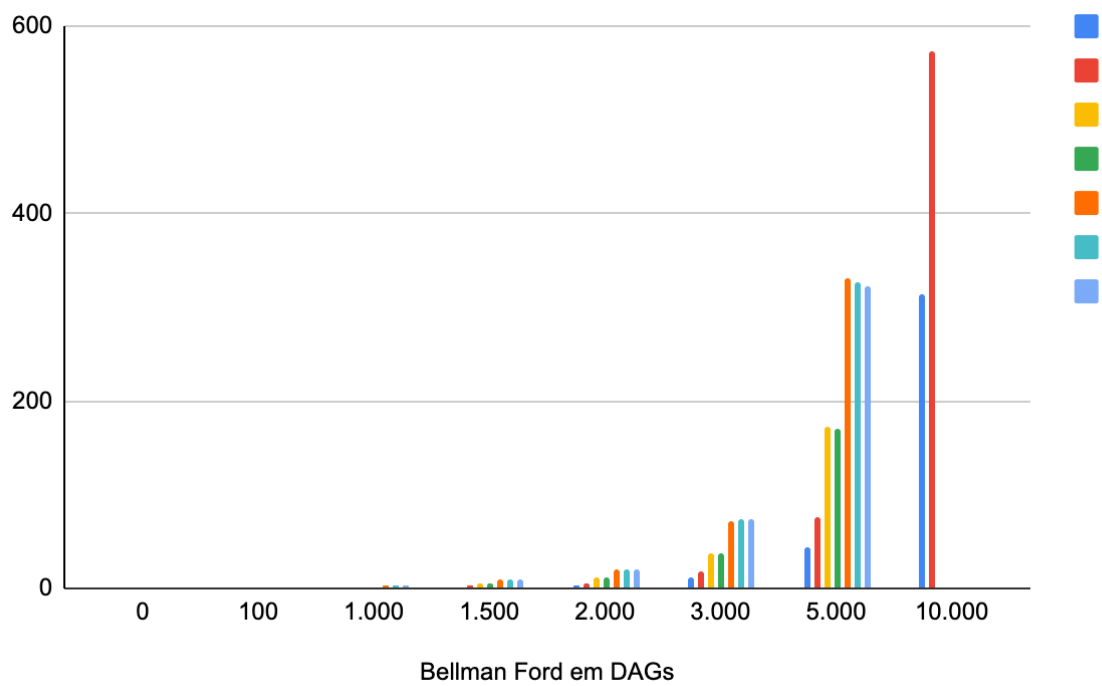
Dijkstra em DAGs

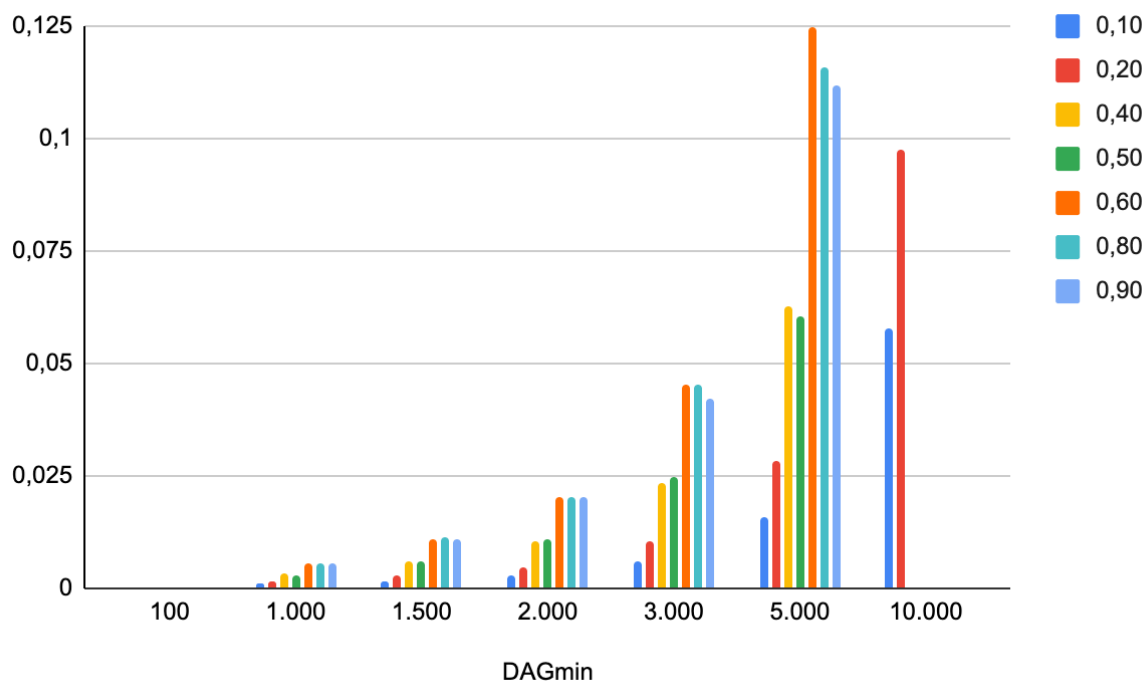
n\p	0,10	0,20	0,40	0,50	0,60	0,80	0,90
100	0,000048	0,000088	0,000109	0,000111	0,000201	0,000202	0,000222
1.000	0,010216	0,013290	0,018928	0,022285	0,026691	0,025131	0,023284
1.500	0,023453	0,032360	0,062970	0,061564	0,076745	0,077328	0,066544
2.000	0,083240	0,087120	0,089795	0,096729	0,130423	0,145424	0,113297
3.000	0,157910	0,198971	0,264490	0,327634	0,379629	0,353767	0,357939
5.000	0,630548	0,747673	0,921236	0,922373	1,259304	1,253785	1,121102
10.000	3,264898	4,074057	-	-	-	-	-

DAGmin

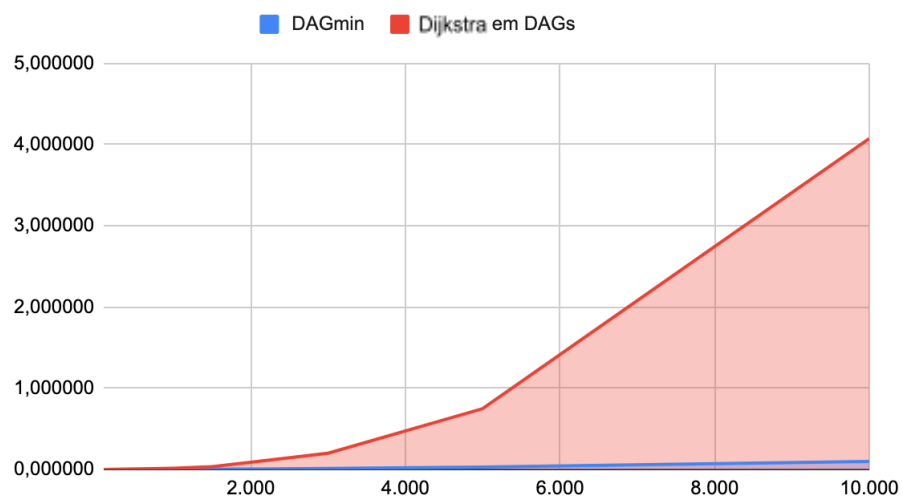
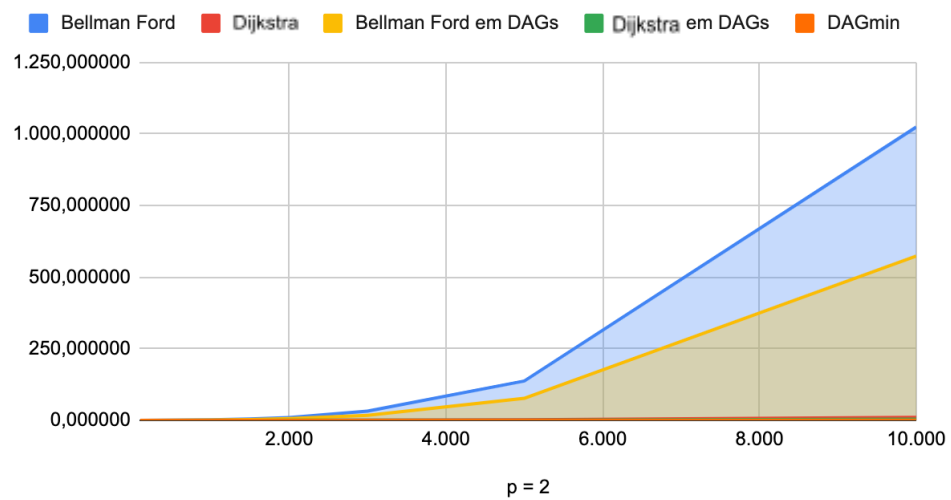
n\p	0,10	0,20	0,40	0,50	0,60	0,80	0,90
100	0,000014	0,000023	0,000028	0,000030	0,000054	0,000049	0,000052
1.000	0,000872	0,001391	0,003321	0,002849	0,005748	0,005509	0,005705
1.500	0,001684	0,002839	0,006214	0,006011	0,010754	0,011518	0,010737
2.000	0,002742	0,004651	0,010650	0,011112	0,020287	0,020077	0,020182
3.000	0,005791	0,010566	0,023273	0,024742	0,045164	0,045440	0,042310
5.000	0,015655	0,028125	0,062569	0,060633	0,124741	0,115956	0,111843
10.000	0,057852	0,097548	-	-	-	-	-







## Algoritmos



A partir da análise dos dados obtidos nos testes realizados, foi possível perceber que, em todos os algoritmos, quanto maior o número de vértices e de arestas maior foi o tempo de execução. Em casos de teste com poucos vértices e arestas a diferença nos tempos de execução não foi significativa o bastante para diferenciar os algoritmos, porém em casos de teste com um maior número de vértices e arestas o algoritmo que limitou os casos de teste foi o Bellman Ford por possuir tempos de execução muito altos para os testes realizados.

O algoritmo Bellman Ford foi o mais lento, apresentando uma notação de  $O(V \cdot A)$ , portanto ao aumentar muito o número de vértices e a densidade do grafo o algoritmo aumentou seu tempo de execução consideravelmente. Para DAGs Bellman Ford também foi o mais lento, apenas sendo mais rápido do que o próprio Bellman Ford em Grafos não DAGs por ter menos arestas que este.

Dijkstra foi consistentemente rápido em todos os casos de teste, tanto em Grafos cíclicos quanto em DAGs, apresenta uma notação de  $O((V+A)\log V)$ .

O algoritmo DAGmin foi ainda mais rápido que o Dijkstra ao analisar DAGs possuindo notação  $O(V+A)$ .