Laboratoare

Simularea si optimizarea arhitecturilor de calcul

Mihai Corneliu-Stefan 244/1

Radulescu Robert-Valentin 244/1

Laborator 2

Cerinte:

b) Lucrare practică: Investigaţii arhitecturale utilizând simulatorul HSA:

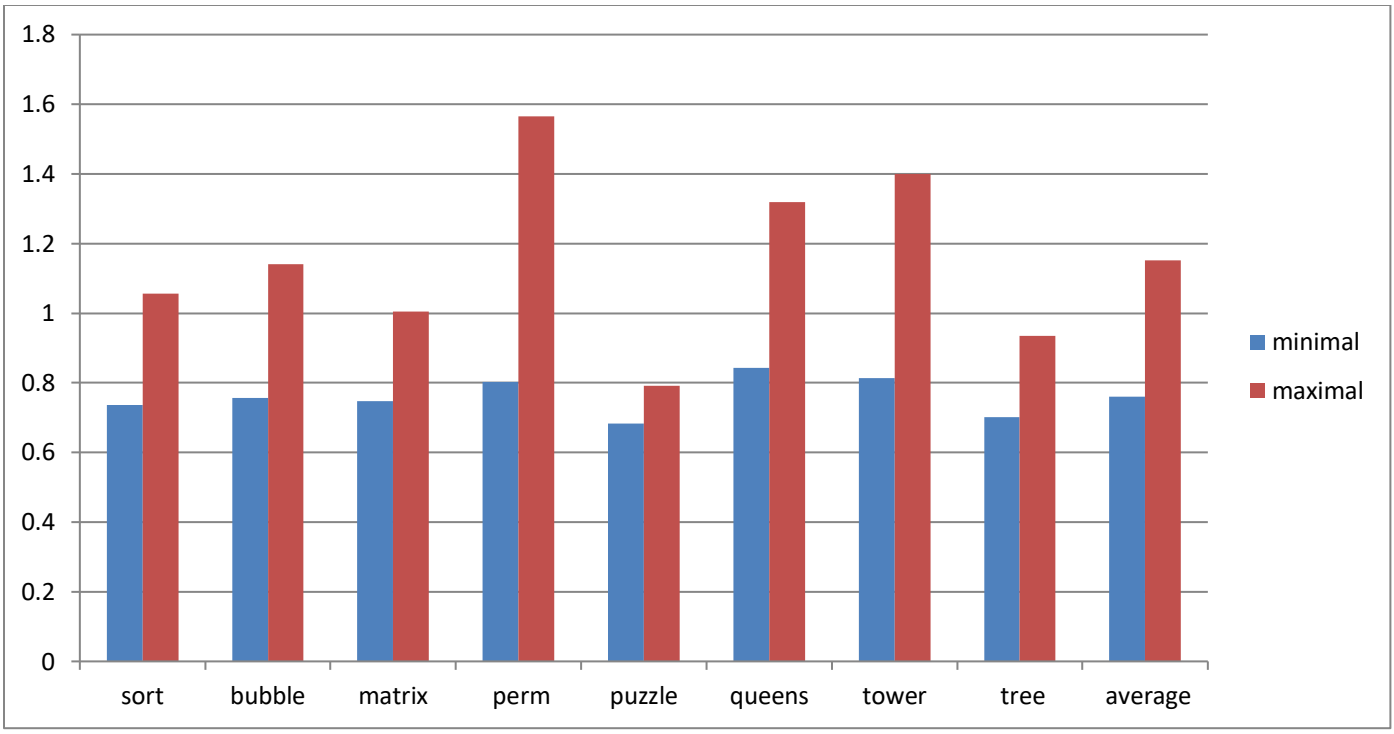
b1) Generaţi în urma simulării de tip *execution driven* fişierele trace (*.trc) aferente celor 8 benchmark-uri Stanford care să cuprindă doar tipurile de instrucţiuni: de salt (*Branch*) şi cele cu referire la memorie (*Load/Store*).

b2) Exemplificaţi grafic rata de procesare obţinută pe cele 8 benchmark-uri în funcţie de tipul modelului arhitectural. **IR = f(**tip_model**)** - *model minimal* vs. *model maximal*. Vizualizaţi gradele de utilizare ale resurselor hardware pe cele 2 modele. Interpretări. Concluzii.

b)

	cicli	instructiuni
sort min	97983	72101
sort max	68203	72101
bubble min	271996	206035
bubble max	180640	206035
matrix min	309973	231814
matrix max	230860	231814
perm min	443567	355643
perm max	227164	355643
puzzle min	10000001	6825651
puzzle max	8614302	6825651
queens min	244991	206420
queens max	156537	206420
tower min	308715	251149
tower max	179478	251149
tree min	193877	136040
tree max	145679	136040

IR	sort	bubble	matrix	perm	puzzle	queens	tower	tree	average
minimal	0.736	0.757	0.748	0.802	0.683	0.843	0.814	0.702	0.760625
maximal	1.057	1.141	1.004	1.566	0.792	1.319	1.399	0.934	1.1515



Din acest exercitiu putem observa ca modelul maximal este mai performant decat modelul minimal din punct de vedere al ratei de procesare. Acest lucru se datoreaza faptului ca modelul maximal poate sa lanseze mai multe instructiuni intr-un ciclu.

Laborator 3

Cerinte:

1. Rezultate urmate de grafice privind influenţa ratei de fetch (FR) asupra ratei de procesare IR(FR) şi asupra ratei de miss în cache-ul de instrucţiuni R_{missIC}(FR).

2. Studiaţi influenţa capacităţii cache-ului de instrucţiuni asupra ratei de procesare IR(SIZE_IC) şi asupra ratei de miss la cache-ul de instrucţiuni R_{missIC}(SIZE_IC).

3. Studiaţi influenţa capacităţii cache-ului de date asupra ratei de procesare IR(SIZE_DC) şi asupra ratei de miss la cache-ul de date R_{missDC}(SIZE_DC).

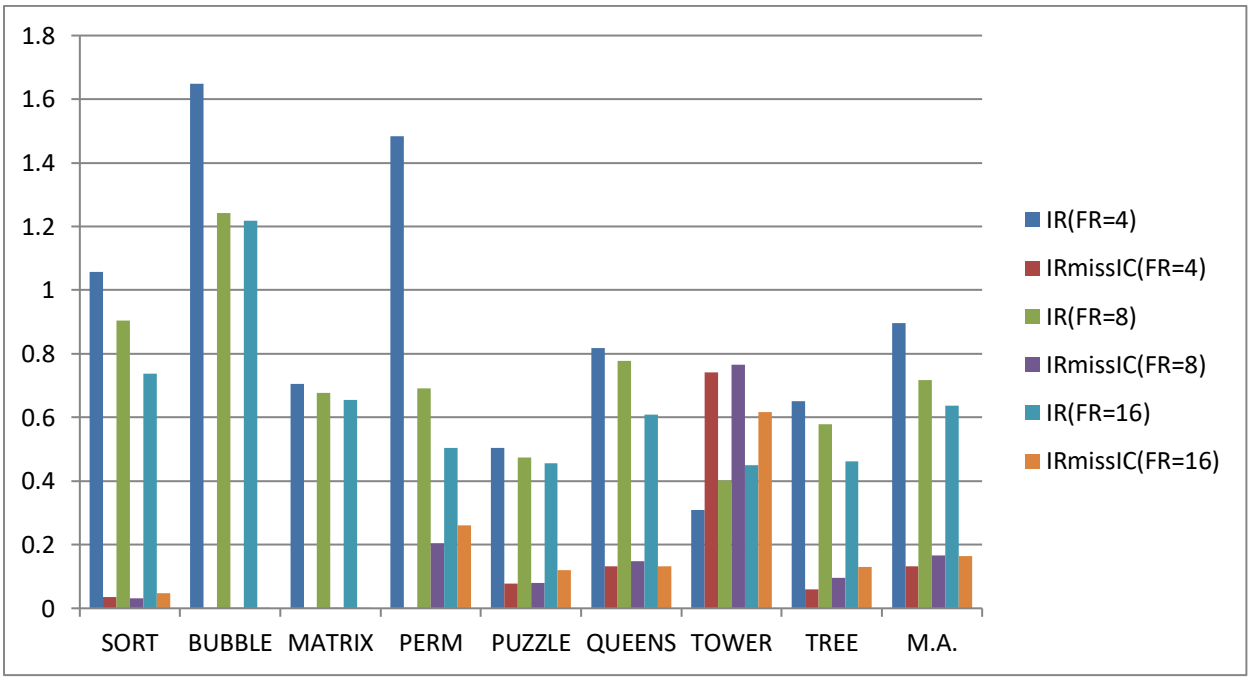
4. Determinaţi influenţa numărului maxim de instrucţiuni ce pot fi trimise simultan în execuţie asupra ratei de procesare IR(IRmax).

5. Se vor genera graficele IR(BLOC_SIZE) şi R_{missDC}(BLOC_SIZE) în cele două ipostaze: scriere în cache prin *write back* şi scriere în cache prin *write through*.

6. Se va studia comparativ realismul, prin rata de procesare, introdus prin cele două tehnici de scriere faţă de situaţia când nu se foloseşte nici una din aceste tehnici IR (tehnica de scriere în cache).

Exercitiul 1

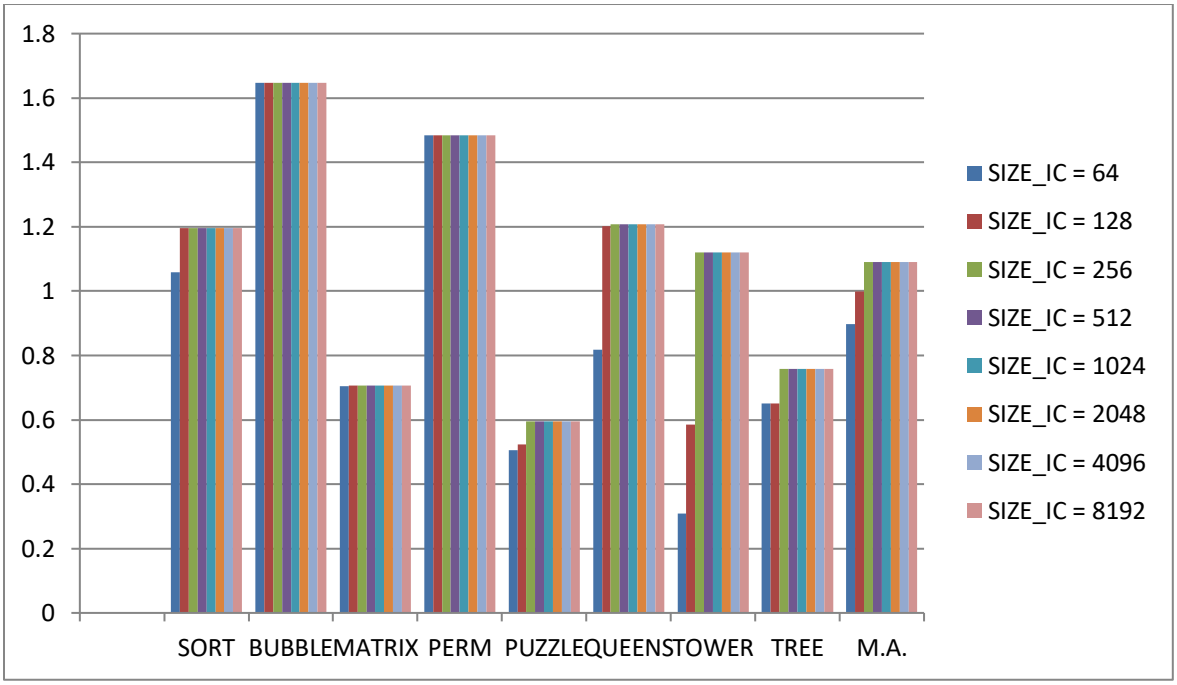
	IR(FR=4)	IRmissIC(FR=4)	IR(FR=8)	IRmissIC(FR=8)	IR(FR=16)	IRmissIC(FR=16)
SORT	1.058	3.50%	0.904	3.09%	0.737	4.70%
BUBBLE	1.648	0.05%	1.242	0.04%	1.218	0.05%
MATRIX	0.705	0.05%	0.678	0.05%	0.654	0.06%
PERM	1.483	0.03%	0.692	20.38%	0.505	26.13%
PUZZLE	0.505	7.79%	0.474	7.94%	0.455	12%
QUEENS	0.818	13.30%	0.777	14.82%	0.608	13.20%
TOWER	0.308	74.04%	0.401	76.46%	0.45	61.70%
TREE	0.651	6.06%	0.578	9.52%	0.462	12.94%
M.A.	0.897	13.102500%	0.71825	16.537500%	0.636125	16.347500%



Din acest exercitiu putem observa ca cel mai mare Issue Rate il obtinem pentru Fetch Rate = 4, iar cel mai mare IC Miss Rate il obtinem pentru Fetch Rate = 16.

Exercitiul 2

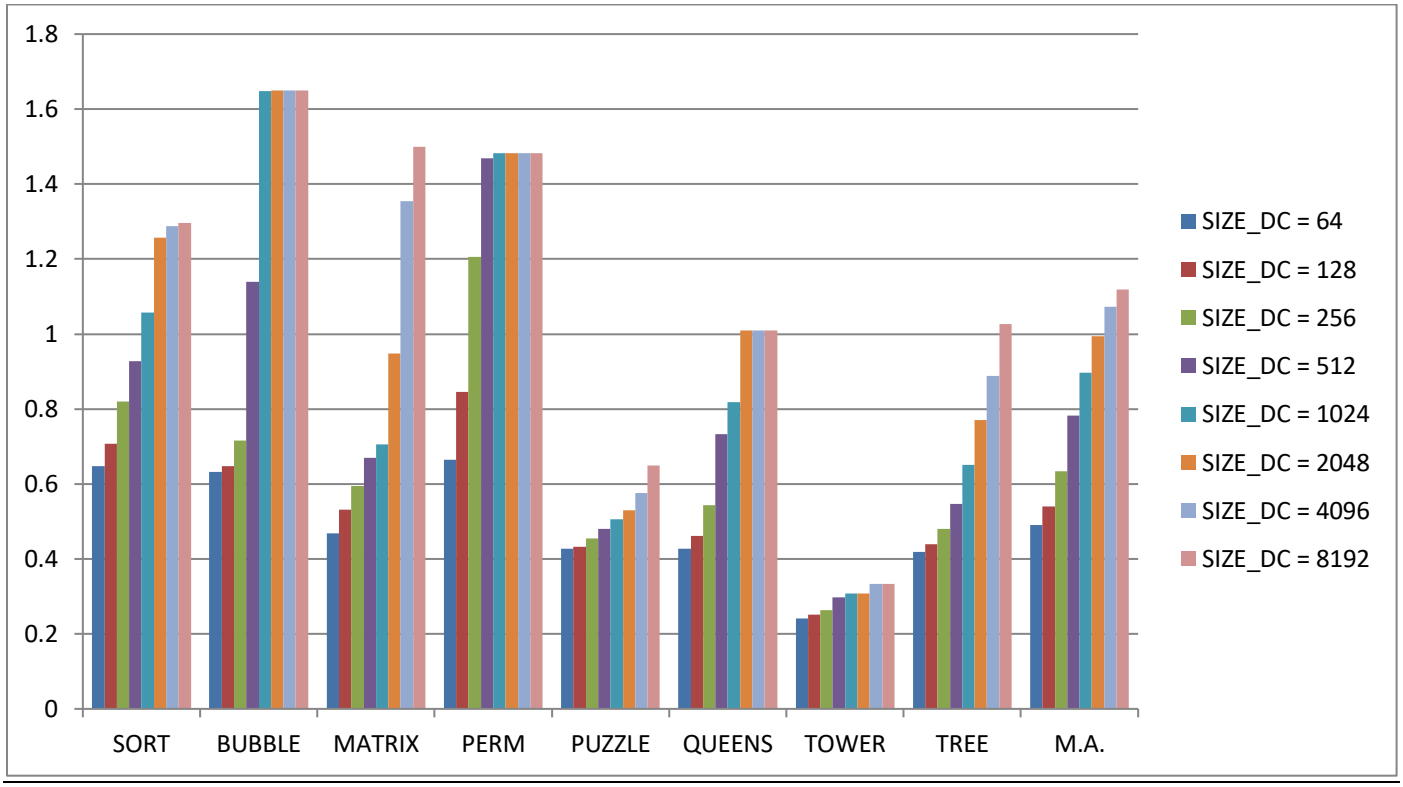
	SIZE_IC = 64	SIZE_IC = 128	SIZE_IC = 256	SIZE_IC = 512	SIZE_IC = 1024	SIZE_IC = 2048	SIZE_IC = 4096	SIZE_IC = 8192
SORT	1.058	1.196	1.196	1.196	1.196	1.196	1.196	1.196
BUBBLE	1.648	1.648	1.648	1.648	1.648	1.648	1.648	1.648
MATRIX	0.705	0.706	0.706	0.706	0.706	0.706	0.706	0.706
PERM	1.483	1.484	1.484	1.484	1.484	1.484	1.484	1.484
PUZZLE	0.505	0.524	0.595	0.595	0.595	0.595	0.595	0.595
QUEENS	0.818	1.201	1.207	1.207	1.207	1.207	1.207	1.207
TOWER	0.308	0.586	1.12	1.12	1.12	1.12	1.12	1.12
TREE	0.651	0.651	0.759	0.759	0.759	0.759	0.759	0.759
M.A.	0.897	0.9995	1.089375	1.089375	1.089375	1.089375	1.089375	1.089375



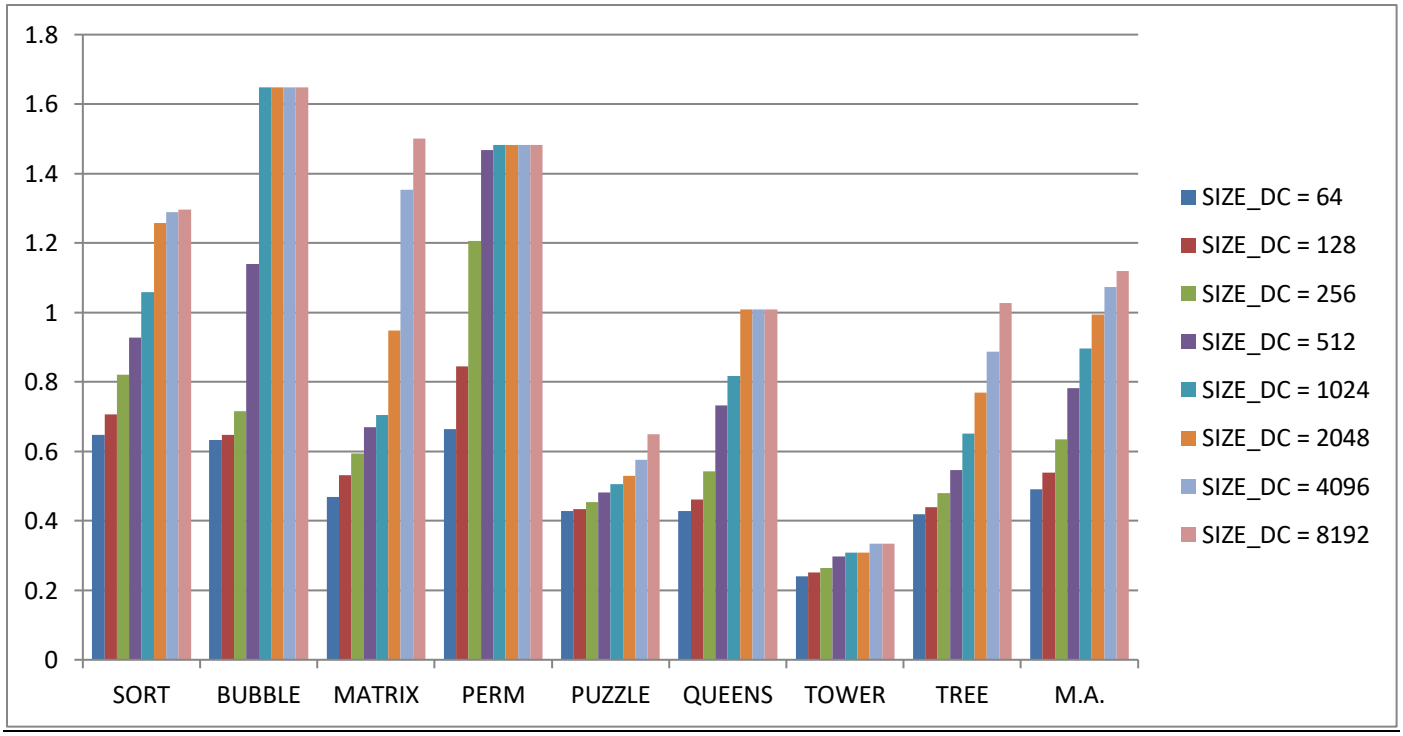
Din acest exercitiu putem observa ca Issue Rate-ul creste pana la SIZE_IC = 256, iar dupa ramane constant. Asadar, nu este rentabil sa folosim un cache mai mare de 256 pentru instructiuni deoarece nu aduce nicio performanta asupra Issue Rate-ului.

Exercitiul 3

	SIZE_DC = 64	SIZE_DC = 128	SIZE_DC = 256	SIZE_DC = 512	SIZE_DC = 1024	SIZE_DC = 2048	SIZE_DC = 4096	SIZE_DC = 8192
SORT	0.648	0.707	0.82	0.928	1.058	1.257	1.288	1.297
BUBBLE	0.632	0.647	0.715	1.139	1.648	1.649	1.649	1.649
MATRIX	0.468	0.532	0.595	0.67	0.705	0.948	1.354	1.5
PERM	0.664	0.845	1.206	1.468	1.483	1.483	1.483	1.483
PUZZLE	0.428	0.433	0.454	0.481	0.505	0.529	0.576	0.649
QUEENS	0.428	0.462	0.543	0.733	0.818	1.009	1.009	1.009
TOWER	0.241	0.252	0.264	0.297	0.308	0.308	0.334	0.334
TREE	0.419	0.44	0.48	0.546	0.651	0.77	0.888	1.027
M.A.	0.491	0.53975	0.634625	0.78275	0.897	0.994125	1.072625	1.1185



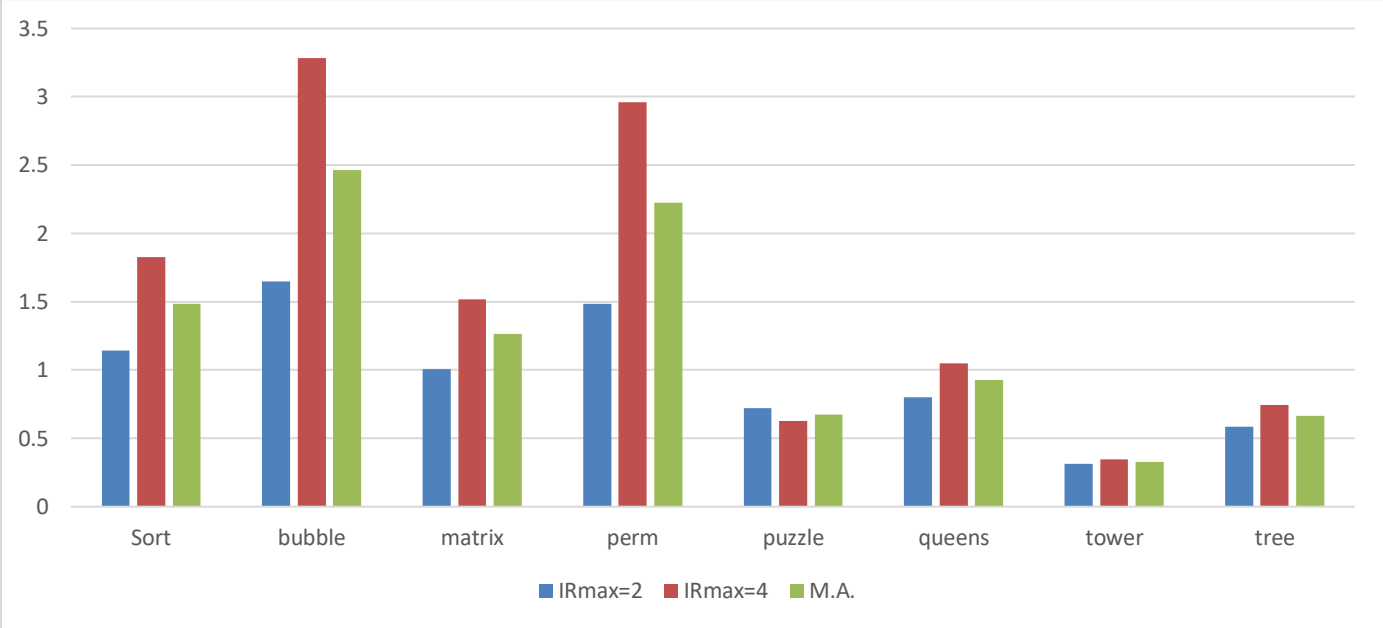
Rata de miss								
	SIZE_DC = 64	SIZE_DC = 128	SIZE_DC = 256	SIZE_DC = 512	SIZE_DC = 1024	SIZE_DC = 2048	SIZE_DC = 4096	SIZE_DC = 8192
SORT	42.52%	36.88%	26.88%	19.05%	12.10%	4.87%	3.76%	3.36%
BUBBLE	33.37%	32.13%	27.18%	9.54%	0.36%	0.34%	0.34%	0.34%
MATRIX	53.76%	44.88%	38.00%	31.42%	28.84%	16.27%	5.32%	2.83%
PERM	38.48%	25.11%	7.64%	0.37%	0.04%	0.03%	0.03%	0.03%
PUZZLE	82.20%	80.62%	74.29%	66.54%	60.21%	53.70%	44.72%	33.16%
QUEENS	65.94%	57.63%	41.59%	20.70%	14.75%	0.22%	0.22%	0.22%
TOWER	54.47%	49.86%	36.17%	14.14%	7.93%	7.55%	0.11%	0.11%
TREE	80.17%	75.41%	65.74%	52.24%	34.54%	20.69%	13.54%	6.44%
M.A.	56.363750%	50.315000%	39.686250%	26.750000%	19.846250%	12.958750%	8.505000%	5.811250%



Din acest exercitiu putem observa ca folosirea unui cache de date cat mai mare creste performanta pentru Issue Rate si scade semnificant Miss Rate in cache-ul de date.

Exercitiul 4

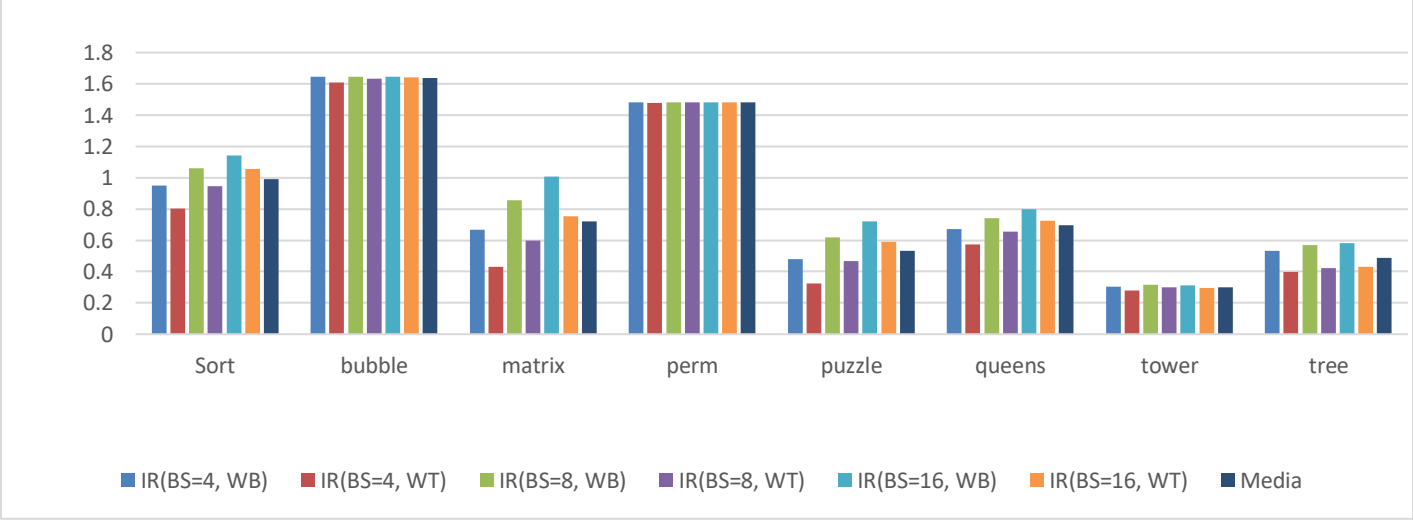
	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
IRmax=2	1.143	1.647	1.009	1.483	0.72	0.8	0.312	0.583
IRmax=4	1.828	3.284	1.517	2.962	0.627	1.051	0.346	0.743
M.A.	1.4855	2.4655	1.263	2.2225	0.6735	0.9255	0.329	0.663



Din acest exercitiu putem observa ca valorile pentru benchmark-uri sunt mai mari atunci cand avem IRmax = 4 comparativ cu IRmax = 2.

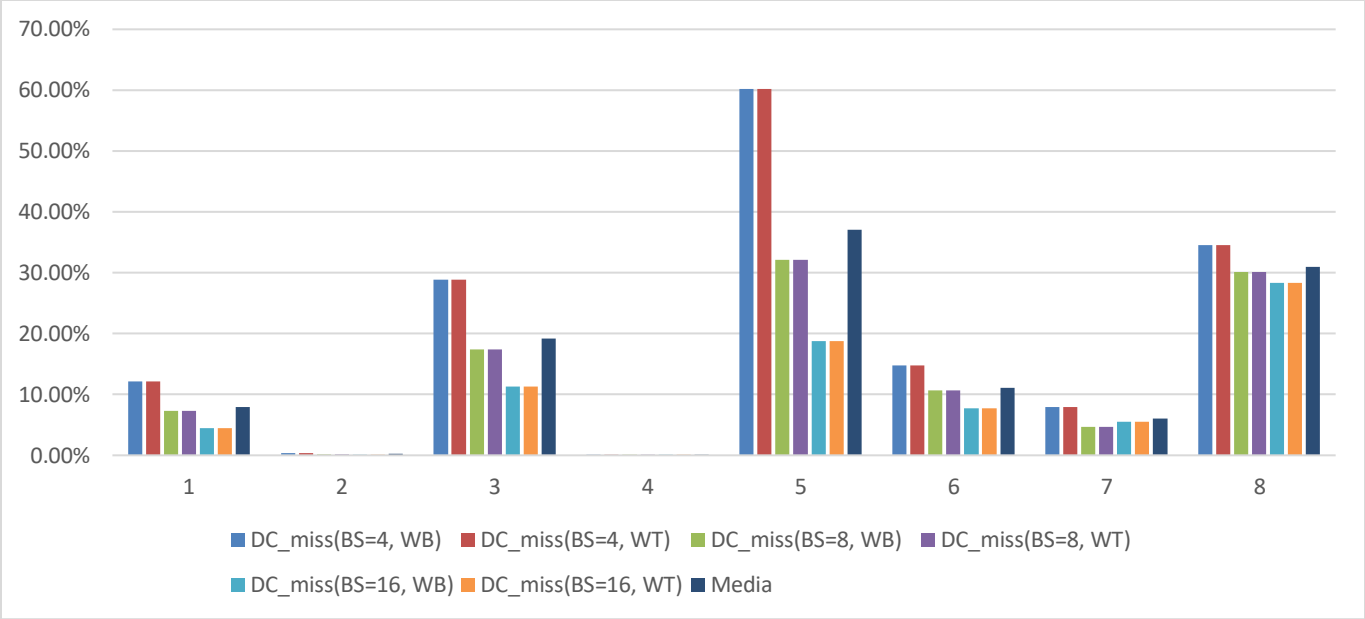
Exercitiul 5

	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
IR(BS=4, WB)	0.949	1.646	0.669	1.483	0.478	0.674	0.303	0.534
IR(BS=4, WT)	0.801	1.61	0.430	1.48	0.323	0.574	0.279	0.398
IR(BS=8, WB)	1.062	1.647	0.858	1.483	0.619	0.742	0.315	0.569
IR(BS=8, WT)	0.945	1.633	0.598	1.481	0.466	0.655	0.299	0.423
IR(BS=16, WB)	1.143	1.647	1.009	1.483	0.720	0.800	0.312	0.583
IR(BS=16, WT)	1.056	1.64	0.753	1.482	0.589	0.726	0.294	0.432
Media	0.992666667	1.637166667	0.7195	1.482	0.5325	0.695167	0.300333	0.489833



Din acest grafic putem observa ca valorile pentru Issue Rate (Write Through si Write Back) sunt crescatoare pe masura ce block size-ul se mareste.

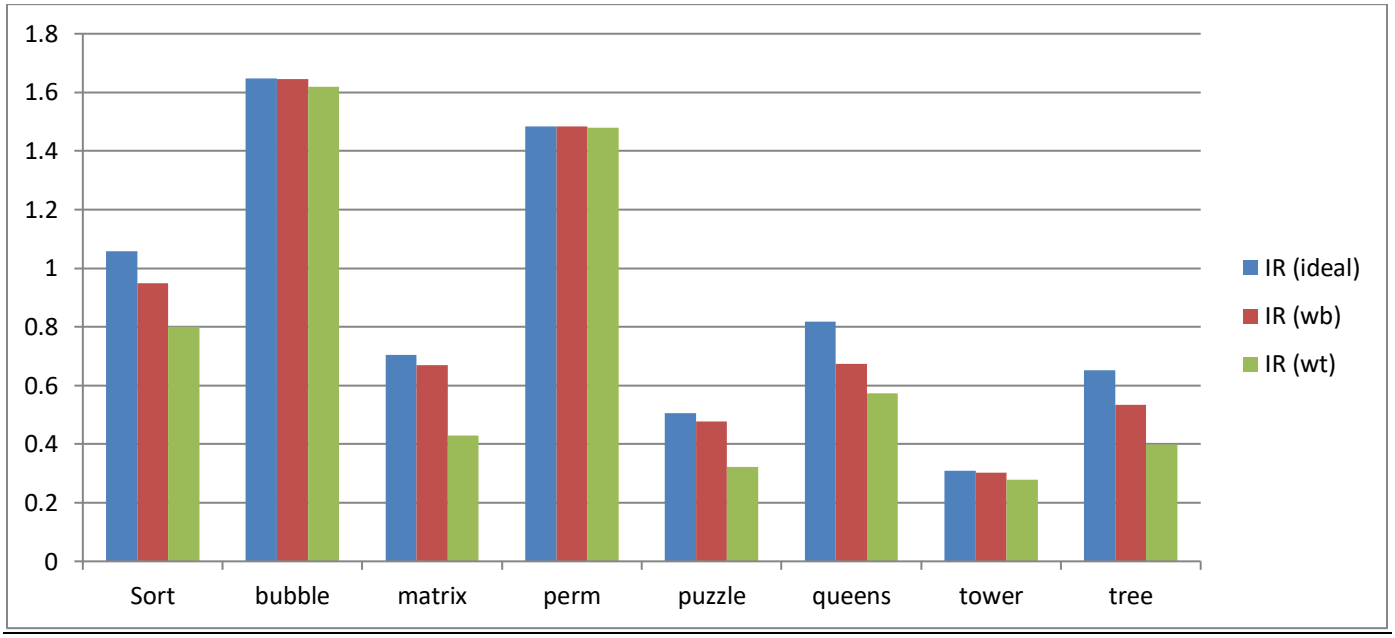
	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
DC_miss(BS=4, WB)	12.10%	0.36%	28.84%	0.04%	60.21%	14.75%	7.93%	34.54%
DC_miss(BS=4, WT)	12.10%	0.36%	28.84%	0.04%	60.21%	14.75%	7.93%	34.54%
DC_miss(BS=8, WB)	7.29%	0.19%	17.39%	0.02%	32.16%	10.68%	4.63%	30.14%
DC_miss(BS=8, WT)	7.29%	0.19%	17.39%	0.02%	32.16%	10.68%	4.63%	30.14%
DC_miss(BS=16, WB)	4.46%	0.10%	11.31%	0.01%	18.81%	7.71%	5.55%	28.36%
DC_miss(BS=16, WT)	4.46%	0.10%	11.31%	0.01%	18.81%	7.71%	5.55%	28.36%
Media	0.0795	0.002166667	0.1918	0.000233	0.3706	0.110467	0.060367	0.310133



Din acest grafic putem observa ca valorile pentru Rmiss (Write Through si Write Back) sunt descrescatoare pe masura ce block size-ul se mareste.

Exercitiul 6

	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
IR (ideal)	1.058	1.648	0.705	1.483	0.505	0.818	0.308	0.651
IR (wb)	0.949	1.646	0.669	1.483	0.478	0.674	0.303	0.534
IR (wt)	0.801	1.619	0.43	1.48	0.323	0.574	0.279	0.398



Din acest exercitiu putem observa ca se obtin valori mai mari pentru Issue Rate folosind tehnica Write Back comparativ cu tehnica Write Through. Totusi, varinta cu Issue Rate (Ideal) este mai buna decat ambele deoarece se obtin rate de procesare putin mai mari.

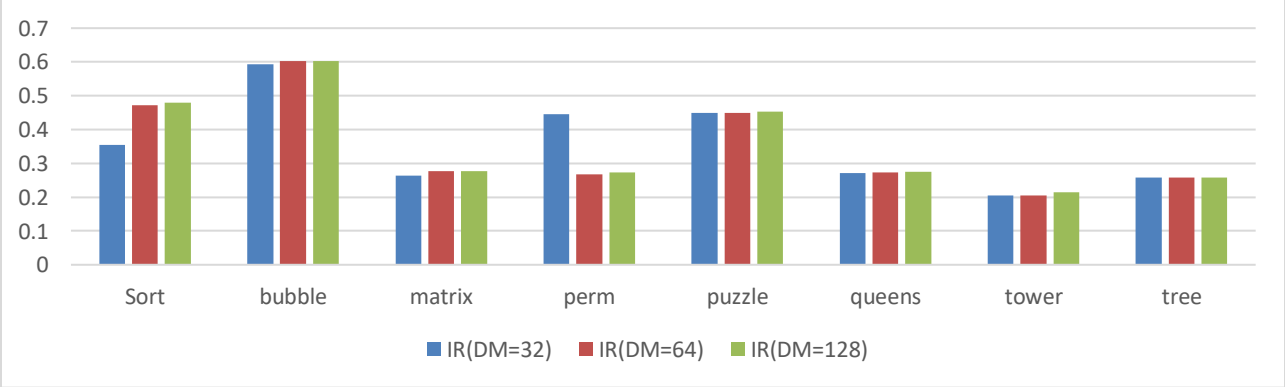
Laborator 4

Cerinte:

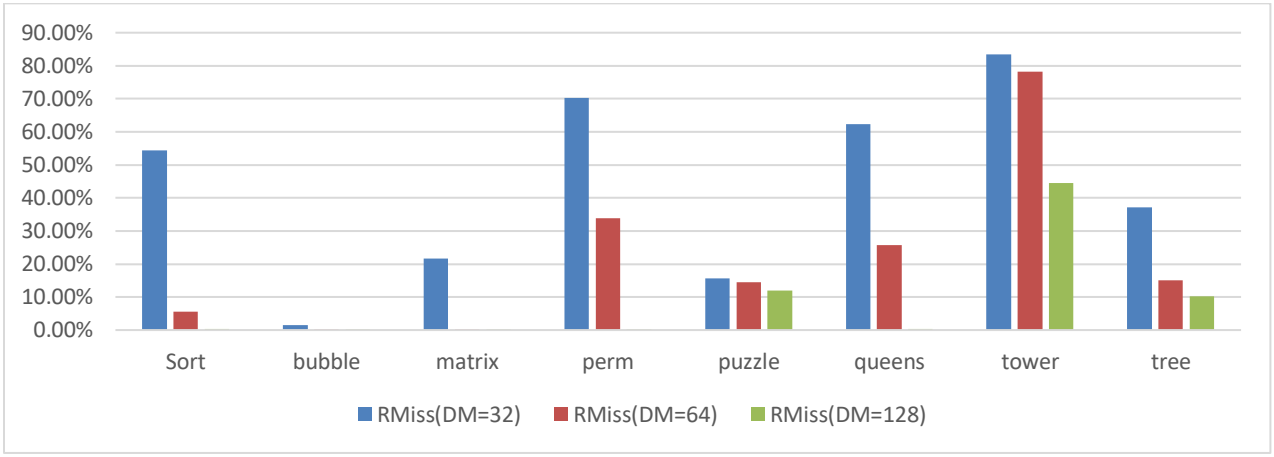
1. Rezultate urmate de grafice privind influenţa capacităţii cache-ului asupra ratei de procesare **IR(DM_size)** şi asupra ratei de miss în cache-ul de instrucţiuni $\mathbf{R_{missIC}}$**(DM_size)** în cele trei situaţii:
   a) fără victim cache.
   b) cu victim cache smplu.
   c) cu selective victim cache.
2. Determinaţi în ce măsură selective victim cache-ul reduce numărul de interschimbări dintre cache-ul principal şi cel victimă **Interchgs(DM_size)** în situaţiile:
   a) cu victim cache smplu.
   b) cu selective victim cache.
3. Studiaţi influenţa capacităţii cache-ului de instrucţiuni asupra ratei de utilizare a respectivului cache **Usage(DM_size)** în situaţiile:
   a) fără victim cache.
   b) cu victim cache smplu.
   c) cu selective victim cache.

Exercitiul 1

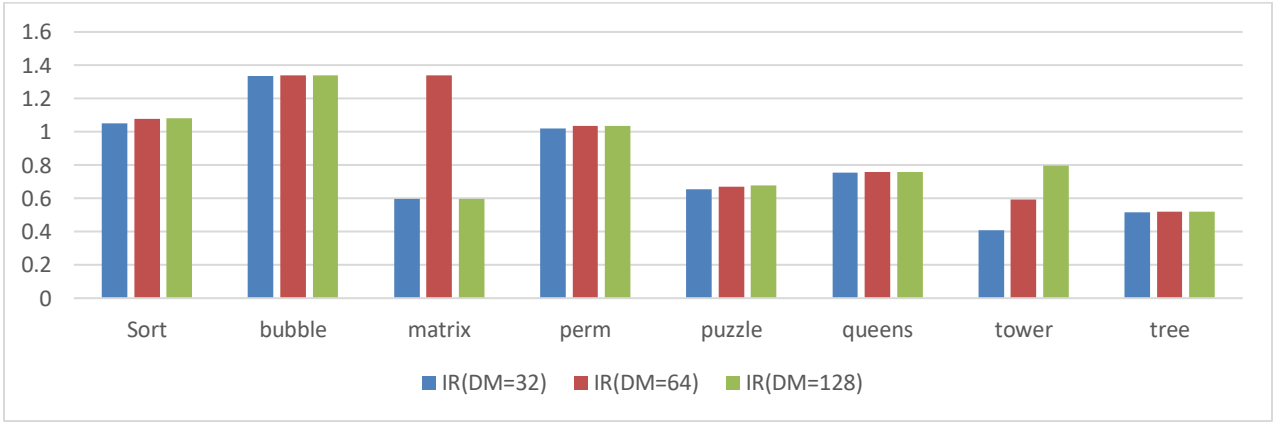
Direct Maped	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
IR(DM=32)	0.3555	0.5931	0.2631	0.4464	0.4502	0.2705	0.205	0.2584
IR(DM=64)	0.4728	0.6024	0.2768	0.2674	0.4502	0.2725	0.205	0.2584
IR(DM=128)	0.4804	0.6024	0.2768	0.2738	0.4538	0.2748	0.2152	0.2589



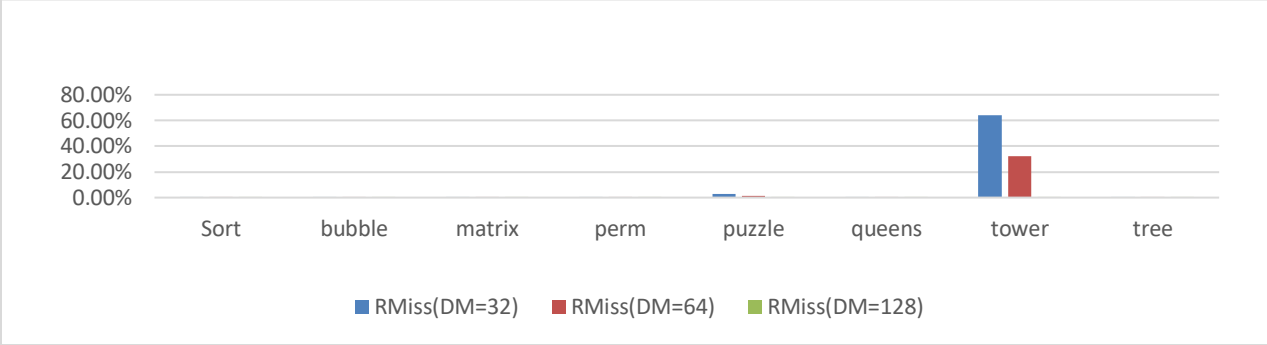
Direct Maped	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
RMiss(DM=32)	54.31%	1.59%	21.57%	70.35%	15.58%	62.40%	83.51%	37.18%
RMiss(DM=64)	5.63%	0.06%	0.11%	33.86%	14.41%	25.66%	78.27%	14.99%
RMiss(DM=128)	0.27%	0.06%	0.09%	0.03%	11.93%	0.39%	44.44%	10.29%



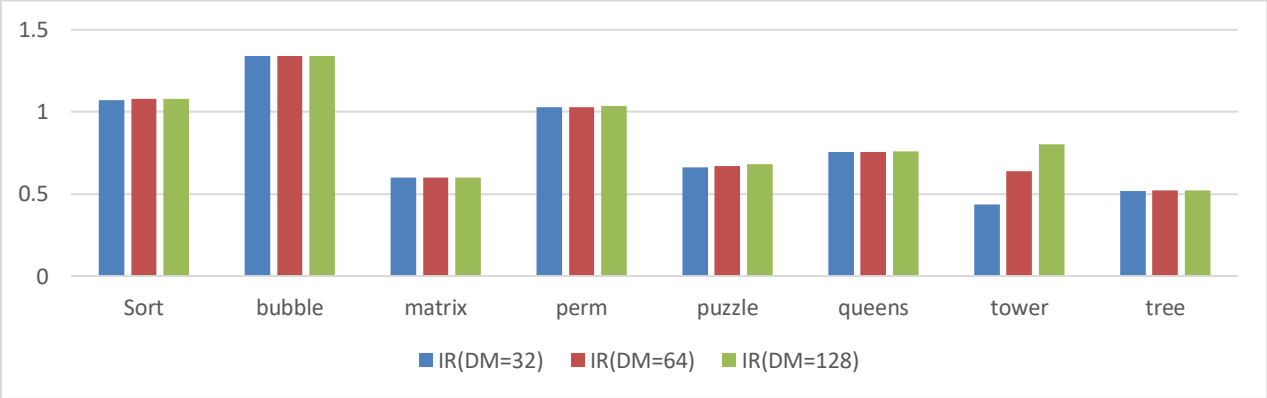
Simple Victim	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
IR(DM=32)	1.0501	1.3337	0.5952	1.0204	0.6563	0.7564	0.4096	0.5158
IR(DM=64)	1.0795	1.3407	1.3407	1.0354	0.6694	0.7573	0.5914	0.5198
IR(DM=128)	1.0802	1.3407	0.5987	1.0355	0.6786	0.7593	0.7963	0.52



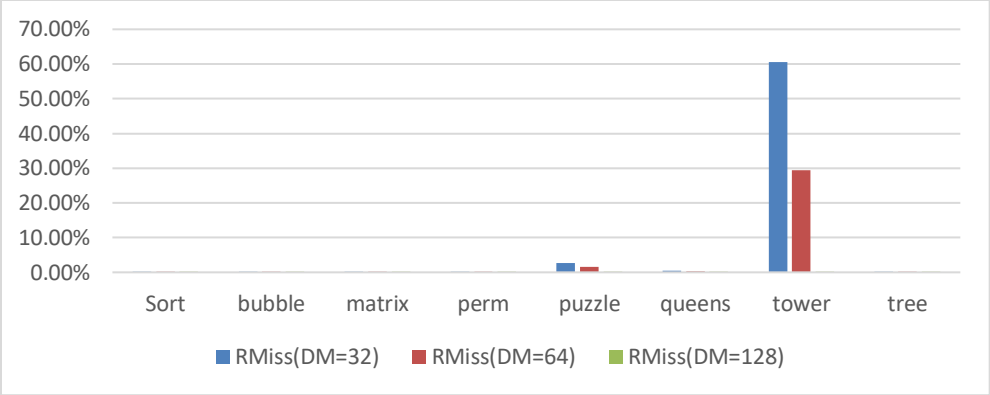
Simple Victim	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
RMiss(DM=32)	0.13%	0.03%	0.04%	0.02%	2.88%	0.49%	64.19%	0.06%
RMiss(DM=64)	0.13%	0.03%	0.04%	0.02%	1.36%	0.37%	32.13%	0.06%
RMiss(DM=128)	0.12%	0.03%	0.04%	0.02%	0.04%	0.05%	0.06%	0.06%



Selective Victim	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
IR(DM=32)	1.0729	1.3388	0.5984	1.0269	0.6625	0.7565	0.4345	0.5187
IR(DM=64)	1.0795	1.3406	0.5987	1.0269	0.6714	0.7574	0.6402	0.5199
IR(DM=128)	1.0801	1.3407	0.5987	1.0355	0.681	0.7593	0.8015	0.52



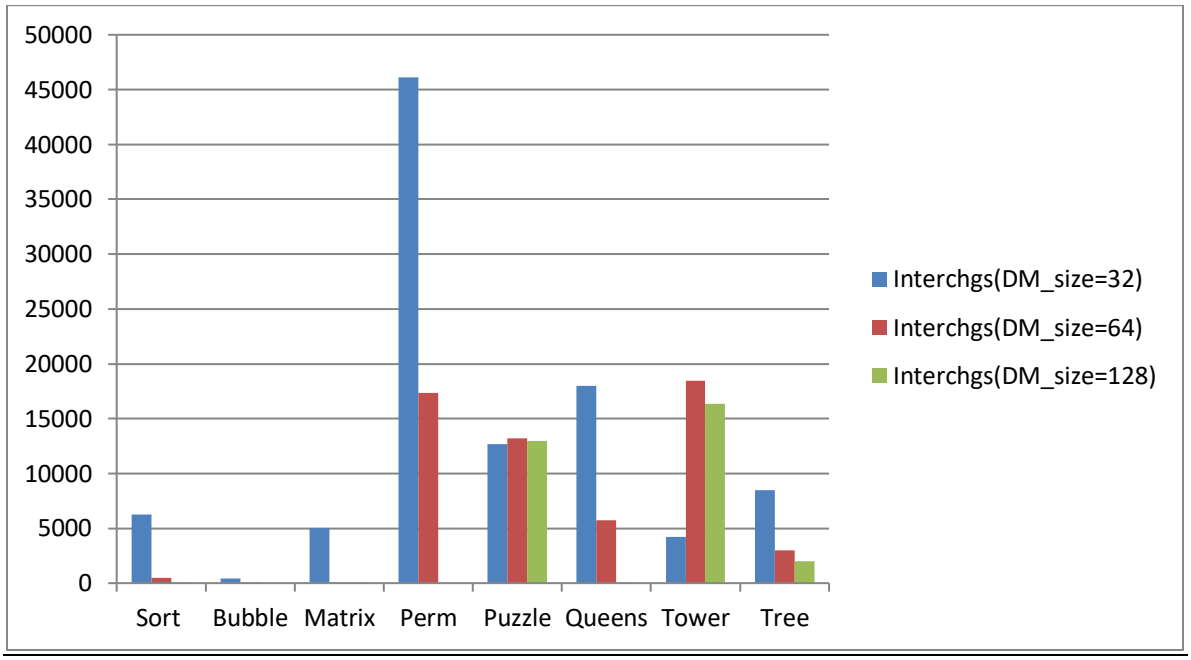
Selective Victim	Sort	bubble	matrix	perm	puzzle	queens	tower	tree
RMiss(DM=32)	0.14%	0.03%	0.04%	0.02%	2.62%	0.49%	60.63%	0.07%
RMiss(DM=64)	0.01%	0.03%	0.04%	0.02%	1.54%	0.33%	29.46%	0.07%
RMiss(DM=128)	0.12%	0.03%	0.04%	0.02%	0.04%	0.05%	0.06%	0.06%



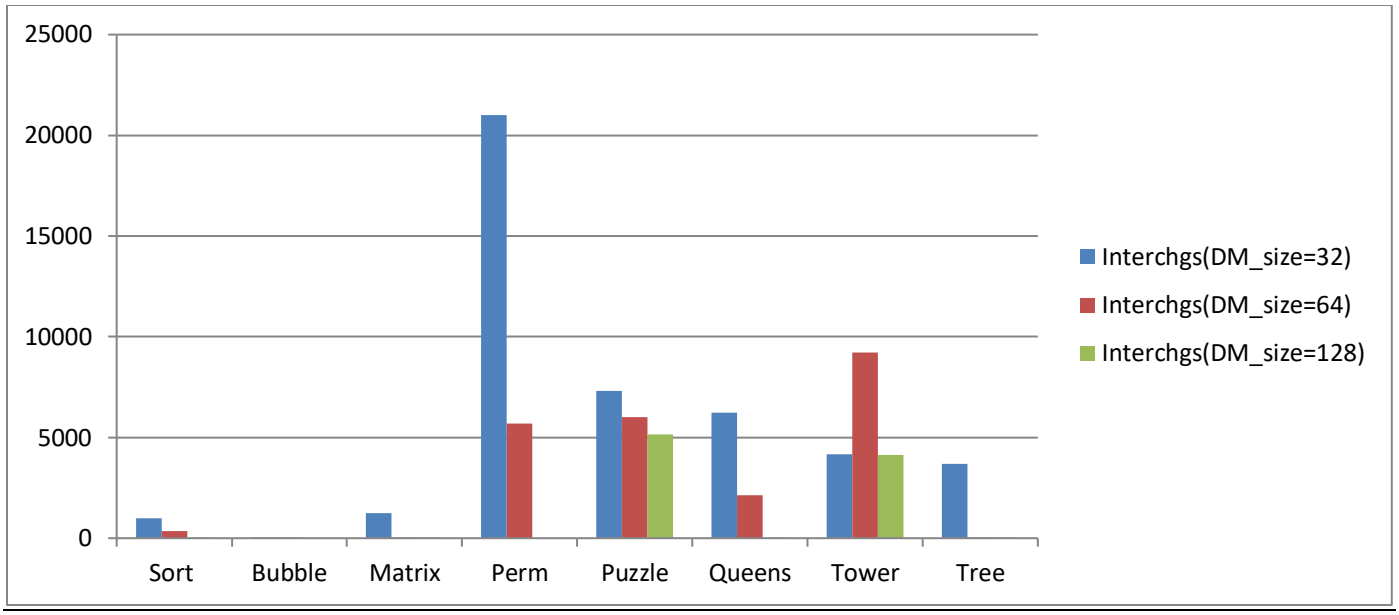
Din acest exercitiu putem observa ca rata de procesare creste cand dimensiunea memoriei de date creste, pe toate cele 3 tipuri de memorie (direct mapped, simple victim, selective victim), insa nesemnificativ. De asemenea, rata de miss scade semnificativ sau in unele cazuri ramane constanta cand dimensiunea memoriei de date creste, pe toate cele 3 tipuri de memorie.

Exercitiul 2

2. a) Victim cache simplu								
	Sort	Bubble	Matrix	Perm	Puzzle	Queens	Tower	Tree
Interchgs(DM_size=32)	6241	399	5049	46135	12685	17971	4224	8477
Interchgs(DM_size=64)	459	2	5	17321	13197	5746	18483	2992
Interchgs(DM_size=128)	2	0	2	0	12992	56	16374	1996



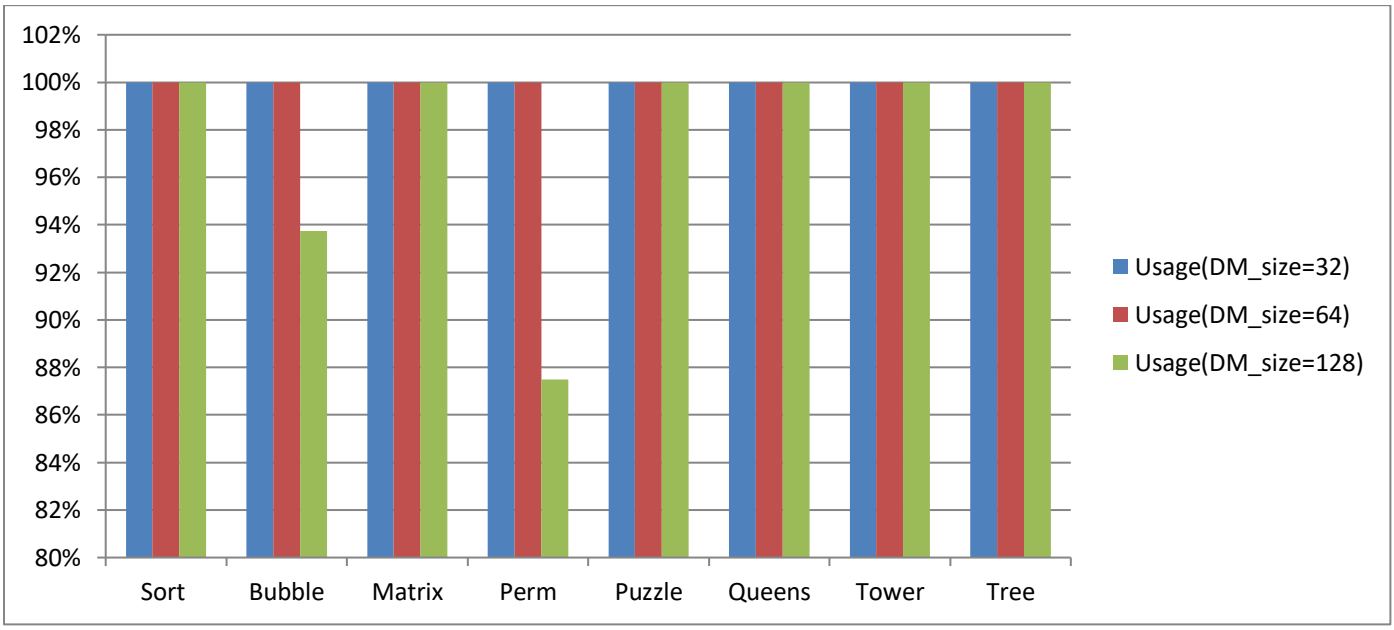
2. b) Selective victim cache								
	Sort	Bubble	Matrix	Perm	Puzzle	Queens	Tower	Tree
Interchgs(DM_size=32)	987	7	1256	21006	7307	6218	4150	3680
Interchgs(DM_size=64)	340	9	11	5682	6006	2137	9223	18
Interchgs(DM_size=128)	5	0	1	0	5162	25	4119	13



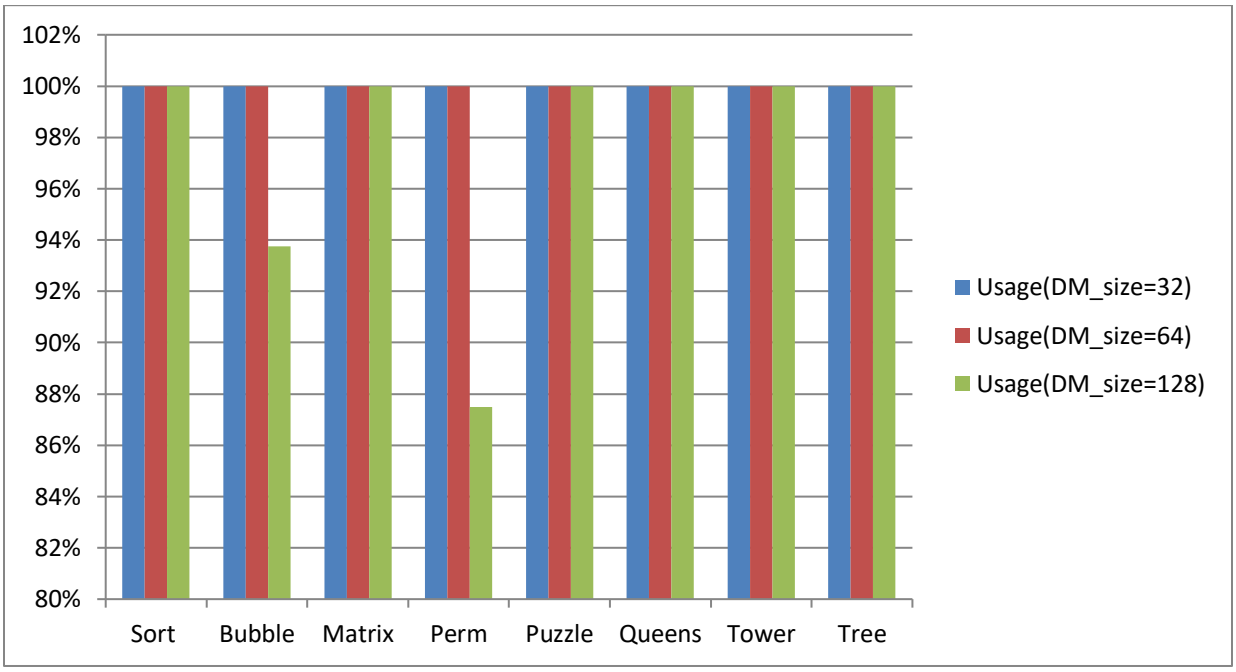
Din acest exercitiu putem observa ca numarul de interschimbari de la victim cache este mai mare decat la selective victim cache. De asemenea, la ambele tipuri de memorie numarul de interschimbari scade cu cat marim dimensiunea cache-ului.

Exercitiul 3

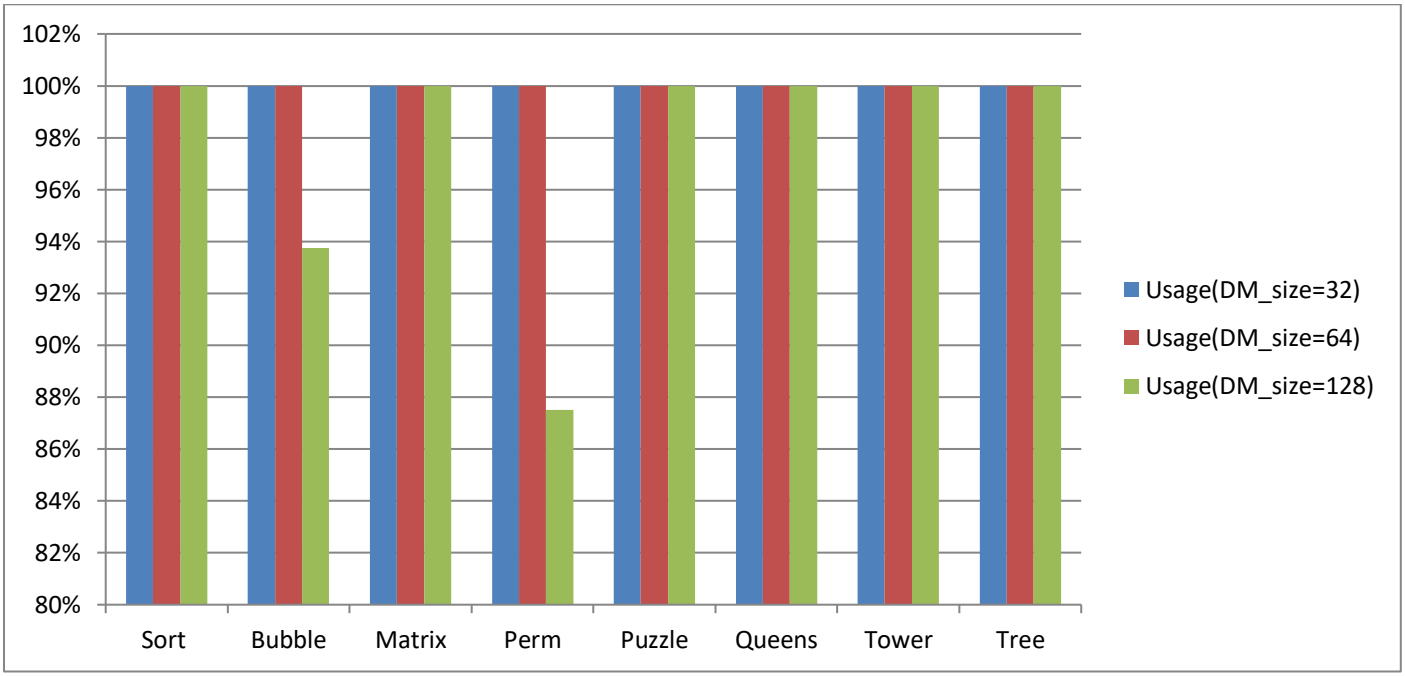
3. a) Fara victim cache								
	Sort	Bubble	Matrix	Perm	Puzzle	Queens	Tower	Tree
Usage(DM_size=32)	100%	100%	100%	100%	100%	100%	100%	100%
Usage(DM_size=64)	100%	100%	100%	100%	100%	100%	100%	100%
Usage(DM_size=128)	100%	93.75%	100%	87.50%	100%	100%	100%	100%



3. b) Cu victim cache simplu								
	Sort	Bubble	Matrix	Perm	Puzzle	Queens	Tower	Tree
Usage(DM_size=32)	100%	100%	100%	100%	100%	100%	100%	100%
Usage(DM_size=64)	100%	100%	100%	100%	100%	100%	100%	100%
Usage(DM_size=128)	100%	93.75%	100%	87.50%	100%	100%	100%	100%



3. c) Cu selective victim cache								
	Sort	Bubble	Matrix	Perm	Puzzle	Queens	Tower	Tree
Usage(DM_size=32)	100%	100%	100%	100%	100%	100%	100%	100%
Usage(DM_size=64)	100%	100%	100%	100%	100%	100%	100%	100%
Usage(DM_size=128)	100%	93.75%	100%	87.50%	100%	100%	100%	100%



Din acest exercitiu putem observa ca rata de procesare ramane constanta atunci cand nu avem victim cache, cand avem victim cache simplu si atunci cand avem selective victim cache.

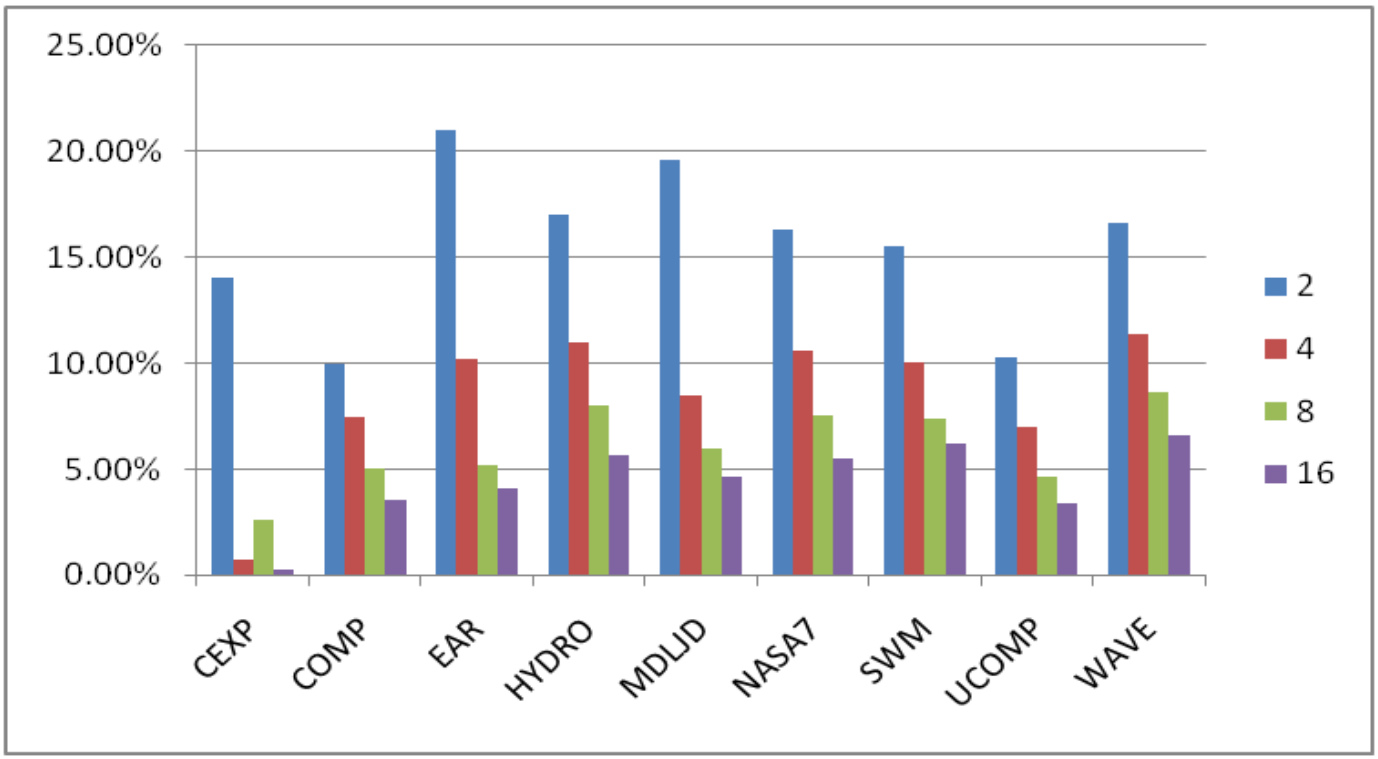
Laborator 5

Cerinte:

1. Incidenta capacitatii cache-ului de instructiuni asupra miss rate
2. Pornind de la configurația inițială generați graficul Rmiss(BLOC_SIZE)
3. Determinati rata de miss variind dimensiunea blocului de date pentru diferite dimensiuni de cache.
4. Determinați rata de miss variind gradul de asociativitate pentru diferite dimensiuni de cache.

Exercitiul 1

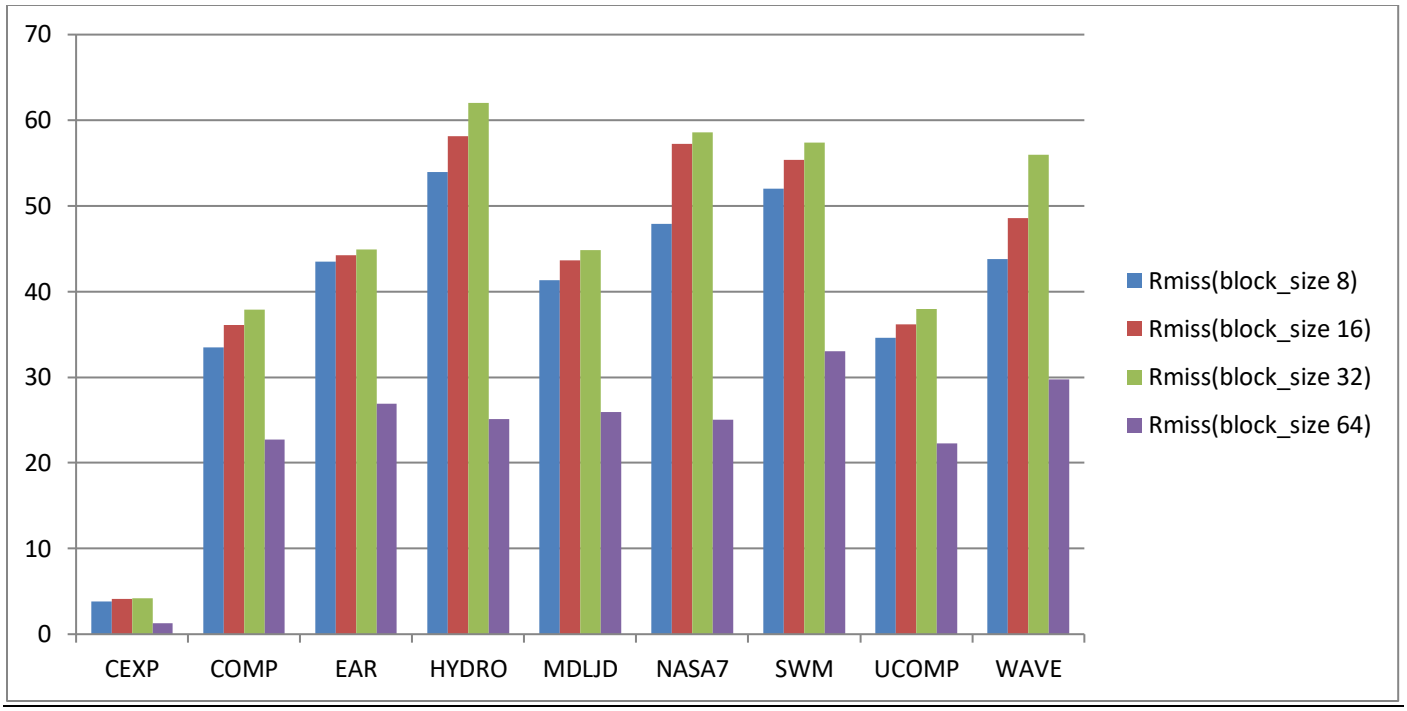
Cache size		2	4	8	16
CEXP		14.05%	0.7%	2.58%	0.21%
COMP		9.94%	7.44%	5.03%	3.56%
EAR		21.04%	10.21%	5.14%	4.1%
HYDRO		16.97%	11%	8.03%	5.68%
MDLJD		19.56%	8.46%	5.94%	4.59%
NASA7		16.28%	10.56%	7.49%	5.5%
SWM		15.53%	10%	7.38%	6.22%
UCOMP		10.24%	6.98%	4.64%	3.4%
WAVE		16.6%	11.4%	8.63%	6.62%



Din acest exercitiu putem observa ca miss rate-ul scade cu cat marim dimensiunea cache-ului.

Exercitiul 2

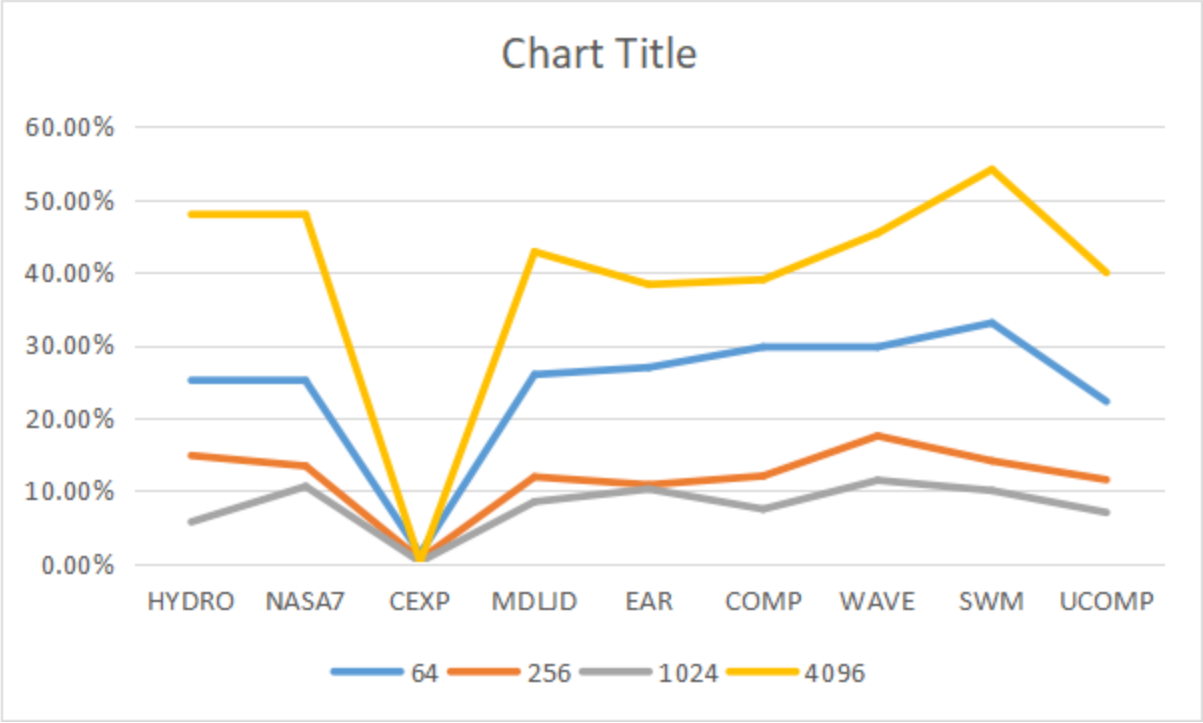
	Rmiss(block_size 8)	Rmiss(block_size 16)	Rmiss(block_size 32)	Rmiss(block_size 64)
CEXP	3.81%	4.14%	4.22%	1.31%
COMP	33.51%	36.13%	37.87%	22.70%
EAR	43.5%	44.23%	44.93%	26.92%
HYDRO	53.97%	58.11%	62.06%	25.15%
MDLJD	41.32%	43.64%	44.82%	25.94%
NASA7	47.87%	57.25%	58.59%	25.06%
SWM	51.99%	55.34%	57.36%	33.04%
UCOMP	34.61%	36.15%	37.94%	22.24%
WAVE	43.79%	48.58%	55.96%	29.73%



Din acest exercitiu putem observa ca Rmiss-ul creste de la block_size = 8 pana la block_size = 32 si scade atunci cand avem block_size = 64.

Exercitiul 3

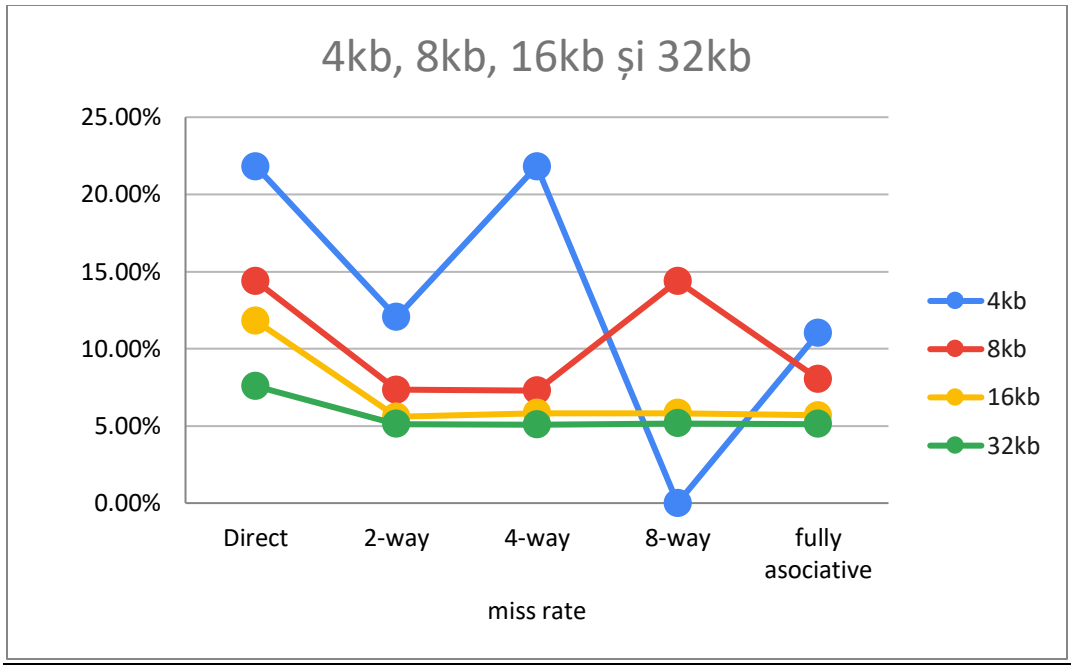
Cache_size = 4kb, bloc_size									
	HYDRO	NASA7	CEXP	MDLJD	EAR	COMP	WAVE	SWM	UCOMP
64	25.15%	**25.07%**	**1.31%**	**25.94%**	**26.92%**	**29.73%**	**29.73%**	**33.04%**	**22.25%**
256	**14.81%**	**13.37%**	**0.55%**	**11.88%**	**10.80%**	**12.01%**	**17.51%**	**14.07%**	**11.49%**
1024	**5.69%**	**10.57%**	**0.22%**	**8.47%**	**10.21%**	**7.45%**	**11.41%**	**10%**	**6.99%**
4096	**47.96%**	**47.87%**	**0.19%**	**42.82%**	**38.32%**	**38.99%**	**45.38%**	**54.15%**	**39.96%**



Din acest exercitiu putem observa ca rata de miss este descrescatoare pentru bloc_size = 64, 256 si 1024 si are parte de o crestere semnificativa atunci cand bloc_size = 4096.

Exercitiul 4

Hydro				
miss rate	4kb	8kb	16kb	32kb
Direct	21.82%	14.39%	11.80%	7.57%
2-way	12.04%	7.33%	5.59%	5.12%
4-way	21.82%	7.29%	5.83%	5.08%
8-way	-	14.39%	5.83%	5.17%
fully asociative	11.00%	8.04%	5.69%	5.12%



Din acest exercitiu putem observa ca rata de miss este cea mai mare pentru cache-ul direct, si cea mai mica per total pentru 2-way associative.

Laborator 6

Cerinte:

1. Sa se reprezinte sub forma grafica functiile utilizând implicit automatul de predictie pe doi biti:
   a) Ap=f(tip_arhitectura)

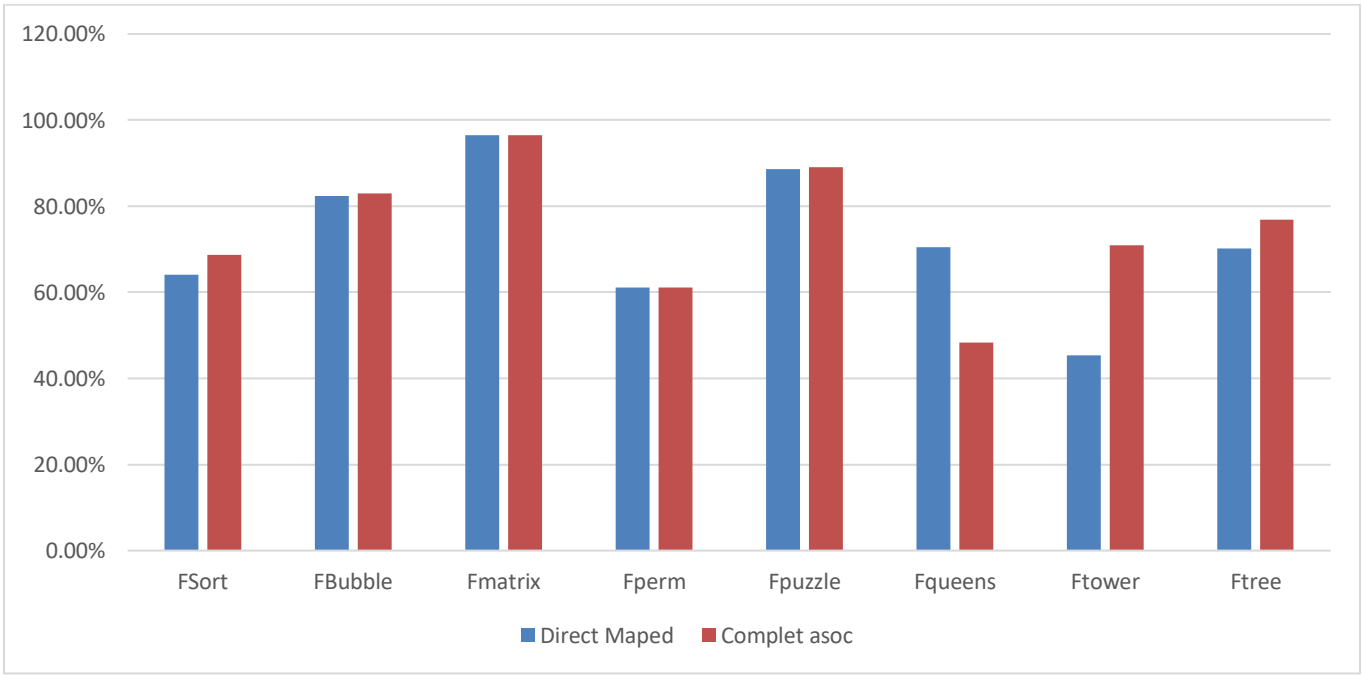
   b)Ap=f(dimensiune_tabela_predictie) Sa se repete rezultatele utilizând automatul de predictie pe 1 bit definit de expresia: ABAB:2.

   c) Reprezentati Ap=f(nr_biti_automat_predictie) considerând parametrul dimensiune_tabela_predictie - valoarea optima rezultata în urma simularii efectuata la b) si arhitectura - optima de la a)

Exercitiul 1

a)

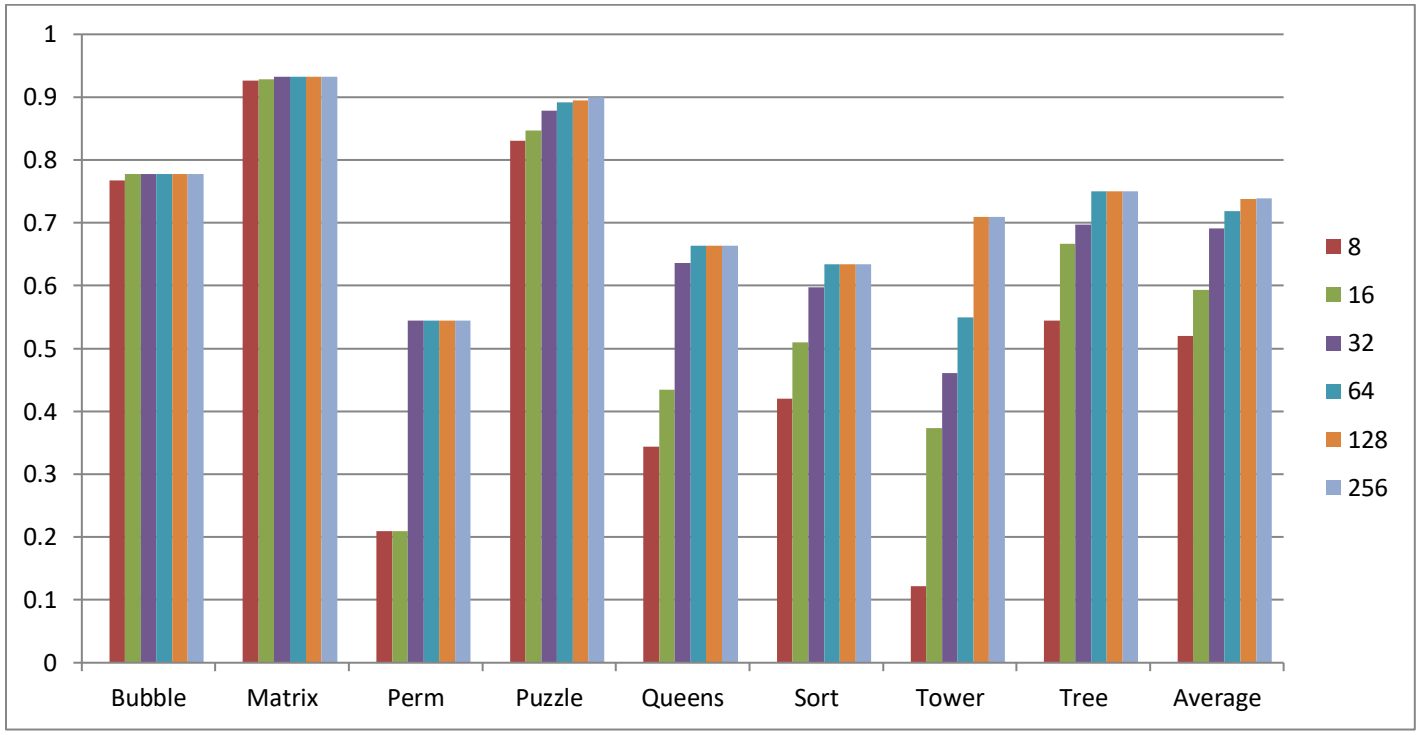
Ap cu btb 32[%]	FSort	FBubble	Fmatrix	Fperm	Fpuzzle	Fqueens	Ftower	Ftree	Average
Direct Maped	64.10%	82.43%	96.53%	61.15%	88.54%	70.54%	45.40%	70.15%	72.355
Complet asoc	68.70%	82.90%	96.53%	61.10%	88.99%	48.29%	70.97%	76.90%	74.2975



Din acest subpunct putem observa ca valorile functiilor utilizand automatul de predictie pe doi biti sunt mai mari la tipul de BTB complet asociativ decat la cel mapat direct.

b)

Mapat										
Numar linii BTB		Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Average
8		76.80%	92.60%	20.90%	83.10%	34.40%	42%	12.20%	54.40%	52.05%
16		77.80%	92.80%	20.90%	84.70%	43.40%	51%	37.30%	66.70%	59.32500%
32		77.80%	93.30%	54.40%	87.90%	63.60%	59.70%	46.10%	69.70%	69.0625%
64		77.80%	93.30%	54.40%	89.20%	66.40%	63.40%	55%	75%	71.812500%
128		77.80%	93.30%	54.40%	89.50%	66.40%	63.40%	70.90%	75%	73.837500%
256		77.80%	93.30%	54.40%	90%	66.40%	63.40%	70.90%	75%	73.900%



Din acest subpunct putem observa ca cele mai mici valori sunt la FTower pentru 8 linii BTB, iar cele mai mari valori sunt la FMatrix de la 32 linii BTB pana la 256 linii, acestea ramanand constante.

c)

Ap		Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Average
1 bit		77.80%	93.30%	54.40%	90%	66.40%	63.40%	70.90%	75%	73.900%
2 biti		82.90%	96.50%	61.10%	91.10%	73.80%	68.70%	70.90%	76.90%	77.737500%

Din acest subpunct putem observa ca valorile automatului de predictie pe 2 biti sunt mai mari decat al celui pe un bit.

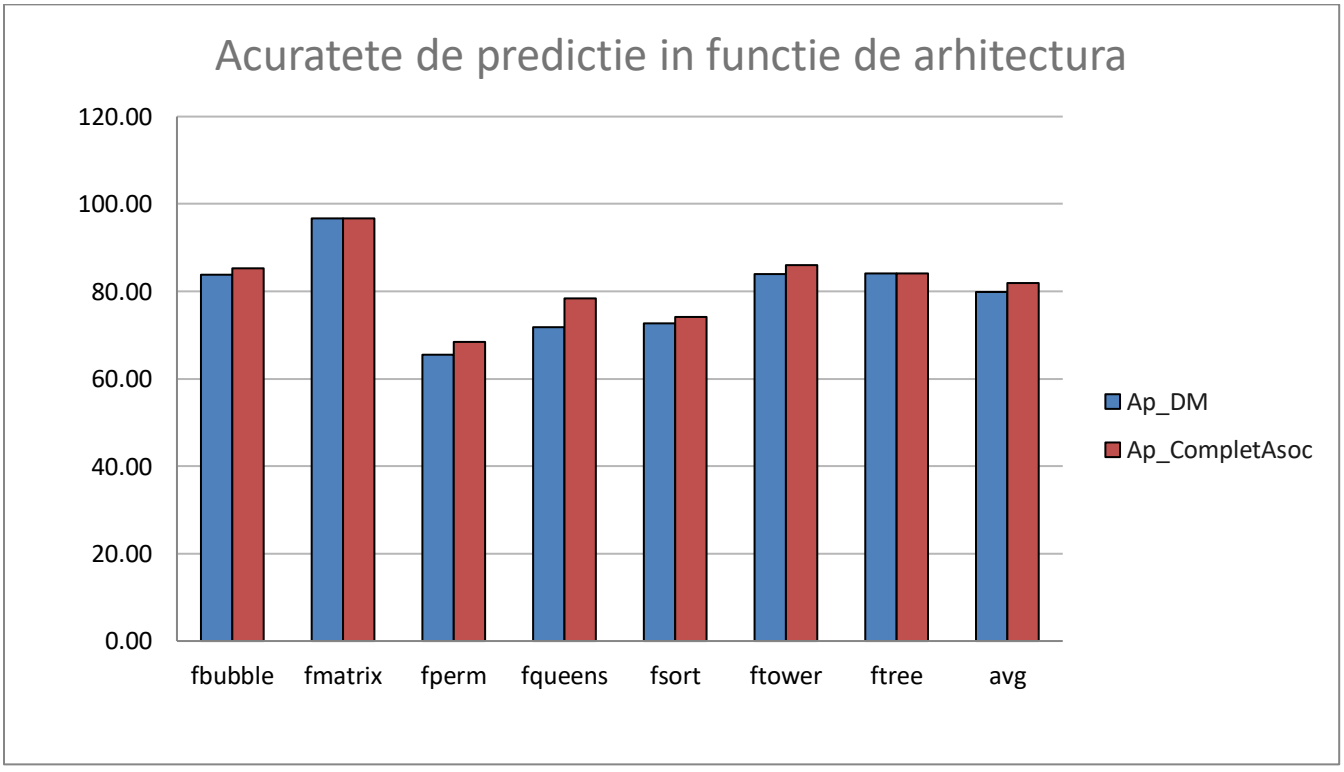
Laborator 7

Cerinte:

1. **Ap=f(tip_arhitectura)**
2. Analizati influenta gradului de localizare al saltului asupra acurateti de predictie: **Ap=f(i)** unde **i** = dimensiunea **PClow.**
3. Stabiliti influenta contextului în care se situeaza saltul în program: **Ap=f(HRglobal)**.
4. Reprezentati **Ap=f(nr_biti_automat_predictie)** considerând parametrii optimi (PClow, HRglobal) rezultati în urma simularii efectuate la 1), 2) si 3).

Exercitiul 1

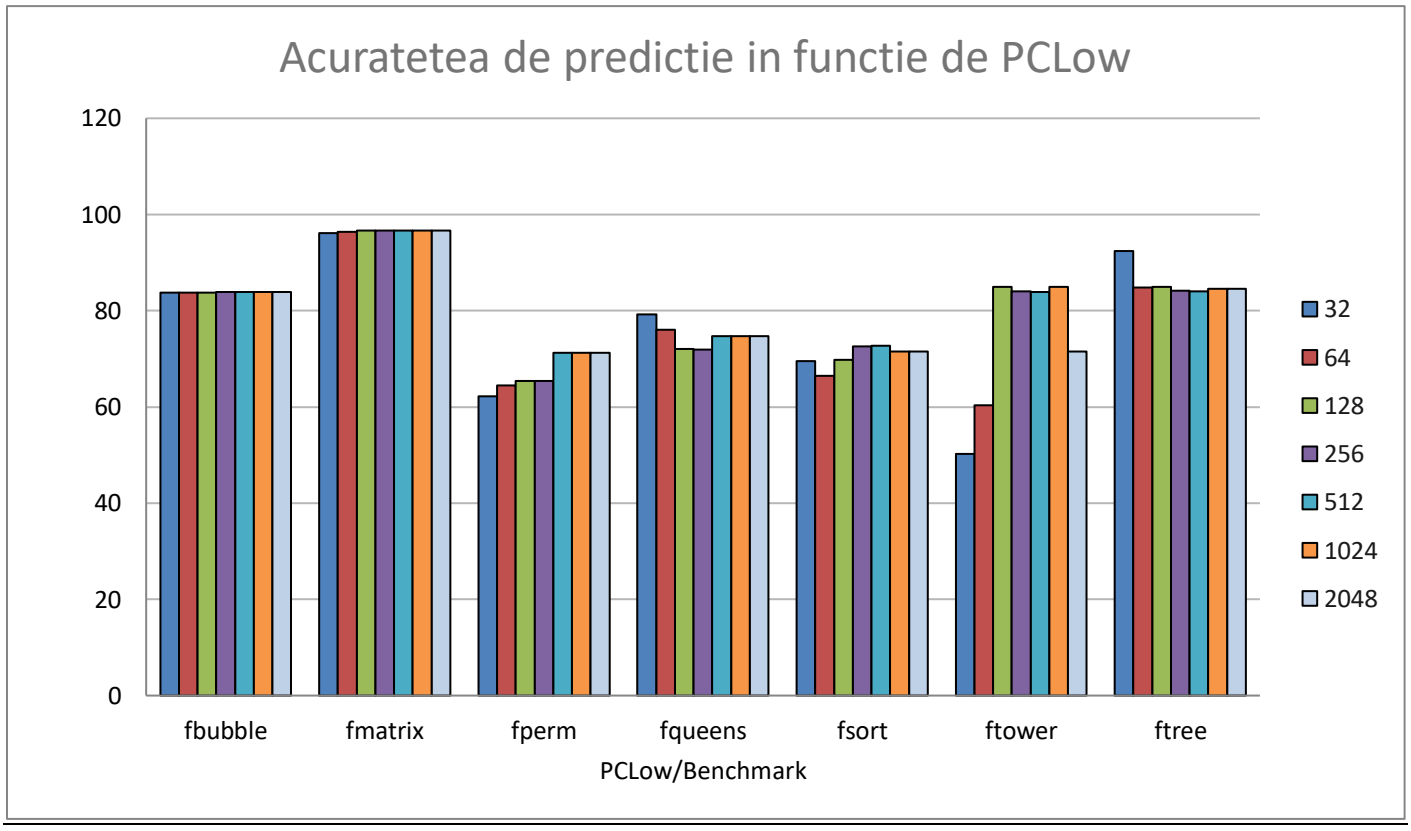
Acuratete de predictie in functie de arhitectura								
	fbubble	fmatrix	fperm	fqueens	fsort	ftower	ftree	avg
Ap_DM	83.86	96.68	65.46	71.86	72.63	84.02	84.10	79.80
Ap_CompletAsoc	85.29	96.68	68.49	78.37	74.15	85.95	84.04	81.85



Din acest exercitiu putem observa ca acuratetea de predictie la arhitectura complet asociativa este mai mare decat la cea mapata direct.

Exercitiul 2

Mapat Direct								
PCLow/Benchmark	fbubble	fmatrix	fperm	fqueens	fsort	ftower	ftree	Average
32	83.7	96.15	62.19	79.27	69.57	50.28	92.38	76.22
64	83.7	96.42	64.53	75.99	66.52	60.4	84.81	76.05285714
128	83.7	96.68	65.46	72.1	69.73	84.91	84.97	79.65
256	83.86	96.68	65.46	71.86	72.63	84.02	84.1	79.80142857
512	83.86	96.68	71.32	74.65	72.71	83.89	84.02	81.01857143
1024	83.86	96.68	71.32	74.77	71.52	84.93	84.5	81.08285714
2048	83.86	96.68	71.32	74.77	71.52	71.52	84.5	79.16714286

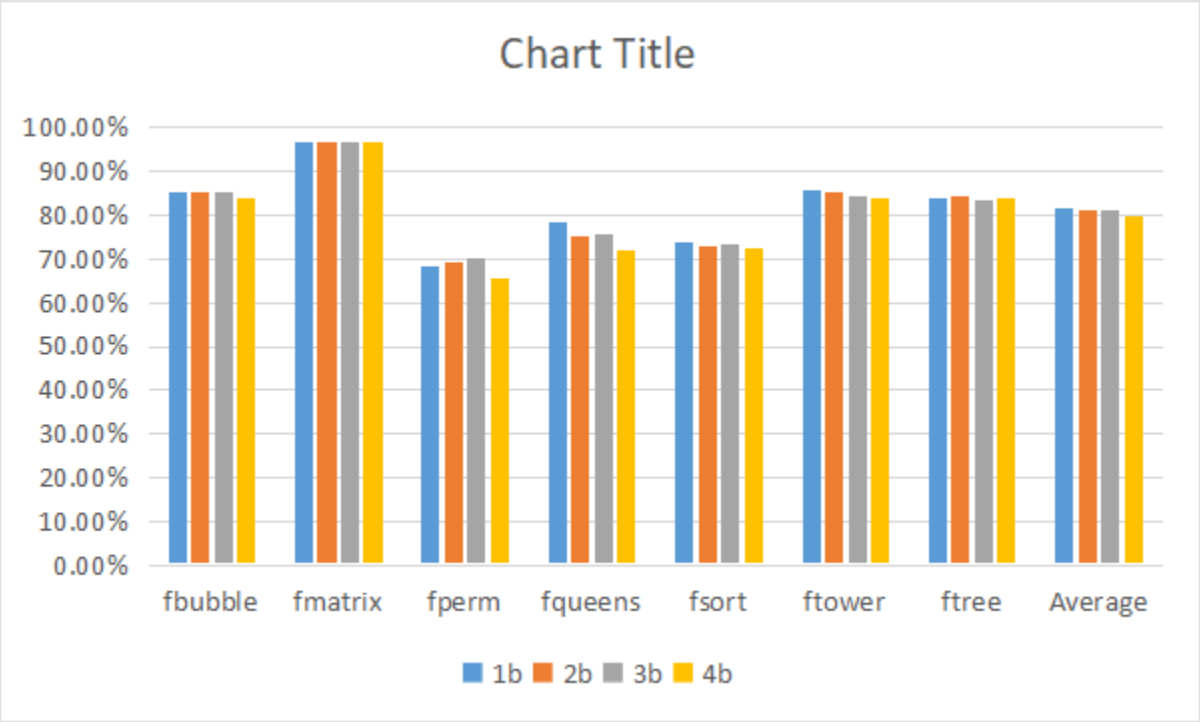


Din acest exercitiu putem observa ca PCLow nu influenteaza semnificativ acuratetea de predictie pentru diferite benchmark-uri, in

unele cazuri aceasta ramanand constanta pentru diferite valori ale lui PCLow.

Exercitiul 3

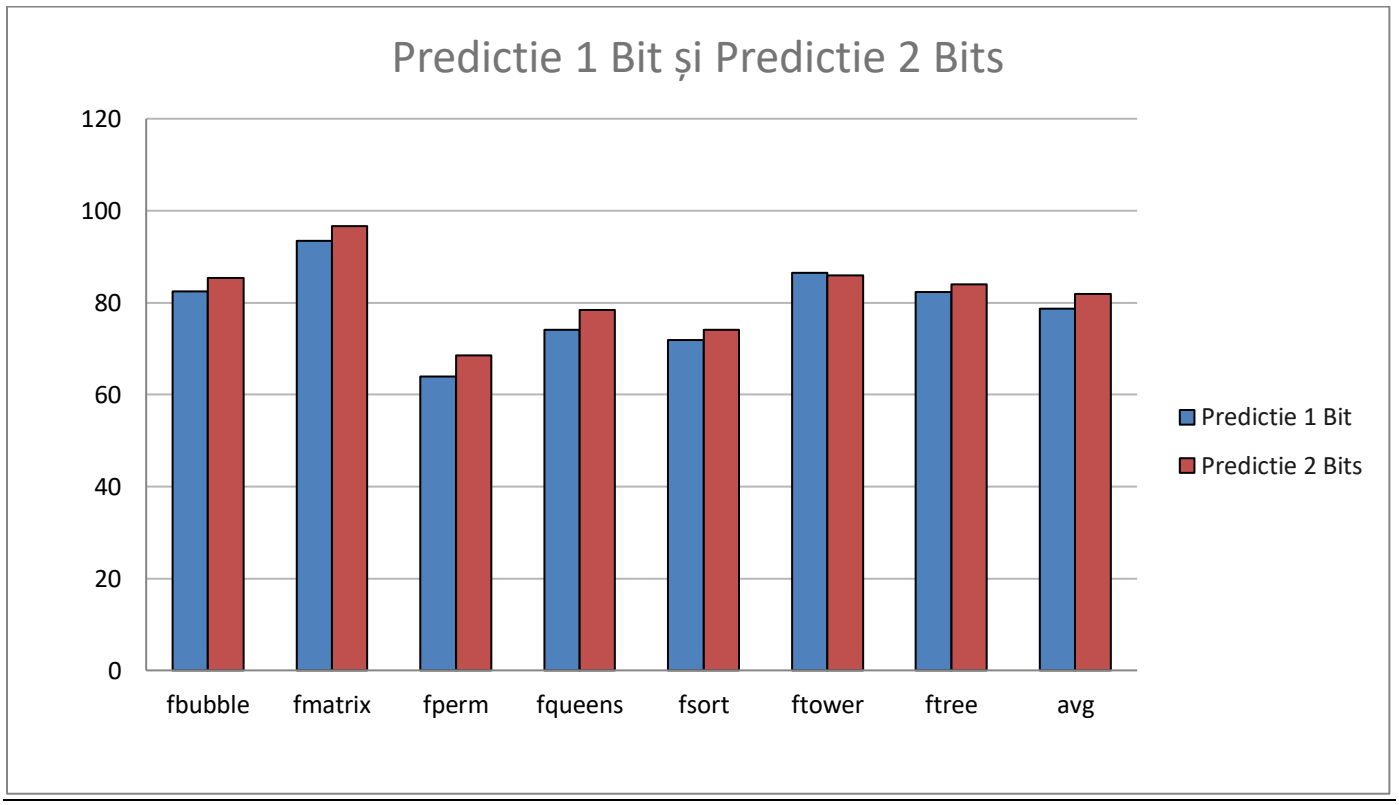
HrGlobal	fbubble	fmatrix	fperm	fqueens	fsort	ftower	ftree	Average
1b	**85.29921%**	**96.6884%**	**68.49995%**	**78.38329%**	**73.66262%**	**85.94981%**	**84.12877%**	**81.80%**
2b	**85.33509%**	**96.6884%**	**69.16988%**	**75.24868%**	**72.74123%**	**85.27139%**	**84.15244%**	**81.23%**
3b	**85.3755%**	**96.6884%**	**70.36263%**	**75.43913%**	**73.34917%**	**84.24799%**	**83.26429%**	**81.25%**
4b	**83.86189%**	**96.68269%**	**65.46281%**	**71.86601%**	**72.63691%**	**84.02491%**	**84.10348%**	**79.81%**



Din acest exercitiu putem observa ca cea mai mare acuratete de predictie o avem pentru 1 bit alocat pentru HrGlobal, iar cea mai mica pentru 4 biti. De asemenea, mediile aritmetice ale benchmark-urilor sunt foarte apropiate pentru toate cele 4 cazuri.

Exercitiul 4

1. Complet Asociativa cu LRU 16								
2. 1024 intrari in tabela								
3. K = 1b								
	fbubble	fmatrix	fperm	fqueens	fsort	ftower	ftree	avg
Predictie 1 Bit	82.45	93.41	63.89	74.09	71.86	86.48	82.32	78.675
Predictie 2 Bits	85.3	96.69	68.5	78.37	74.15	85.95	84.04	81.85714286



Din acest exercitiu putem observa ca acuratetea de predictie este mai mare pentru 2 biti ai predictorului comparativ cu unul. De asemenea, mediile aritmetice sunt apropiate.

Laborator 8

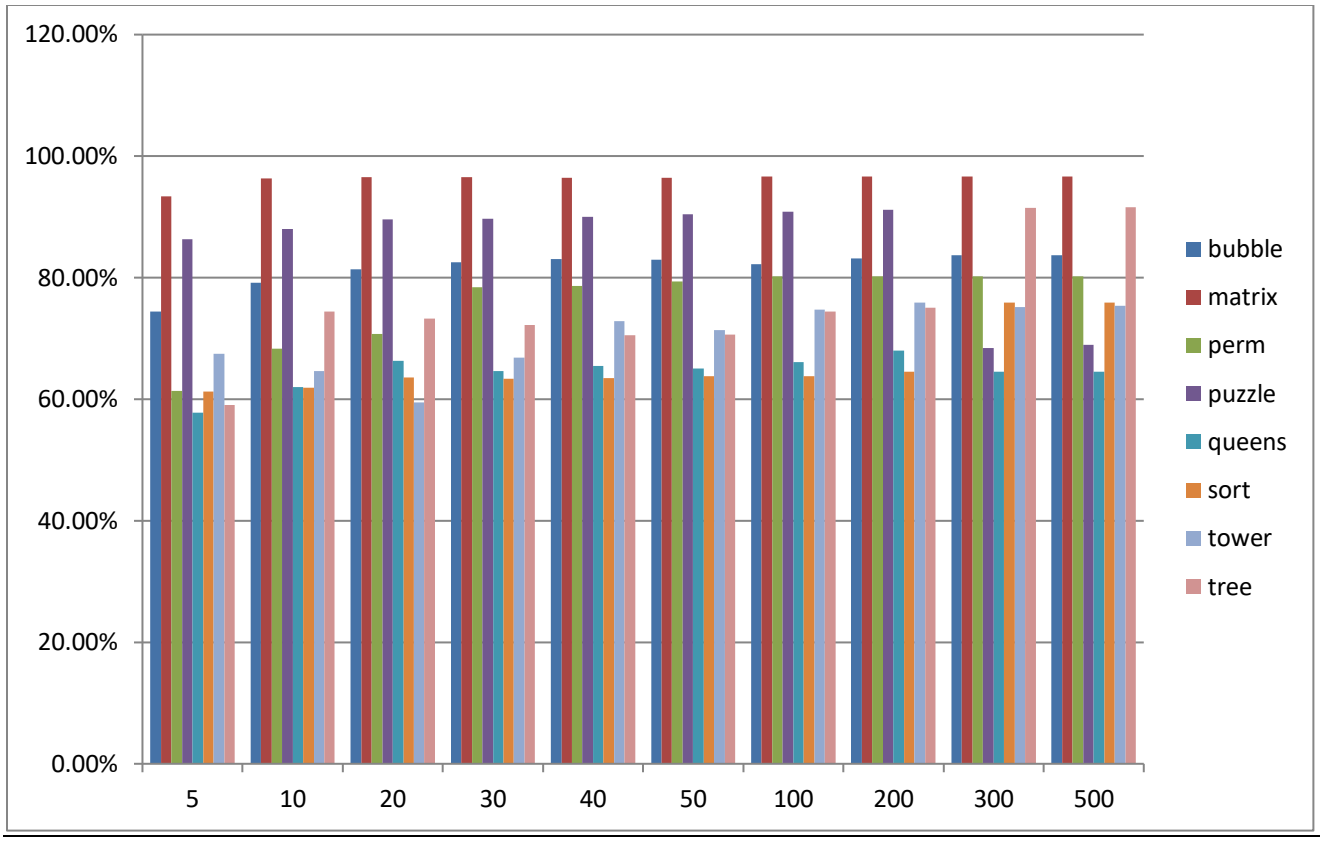
Cerinte:

1. Acuratetea de predictie a simulatorului PPm complet

2. Acuratetea de predictie a simulatorului PPM simplificat

Exercitiul 1

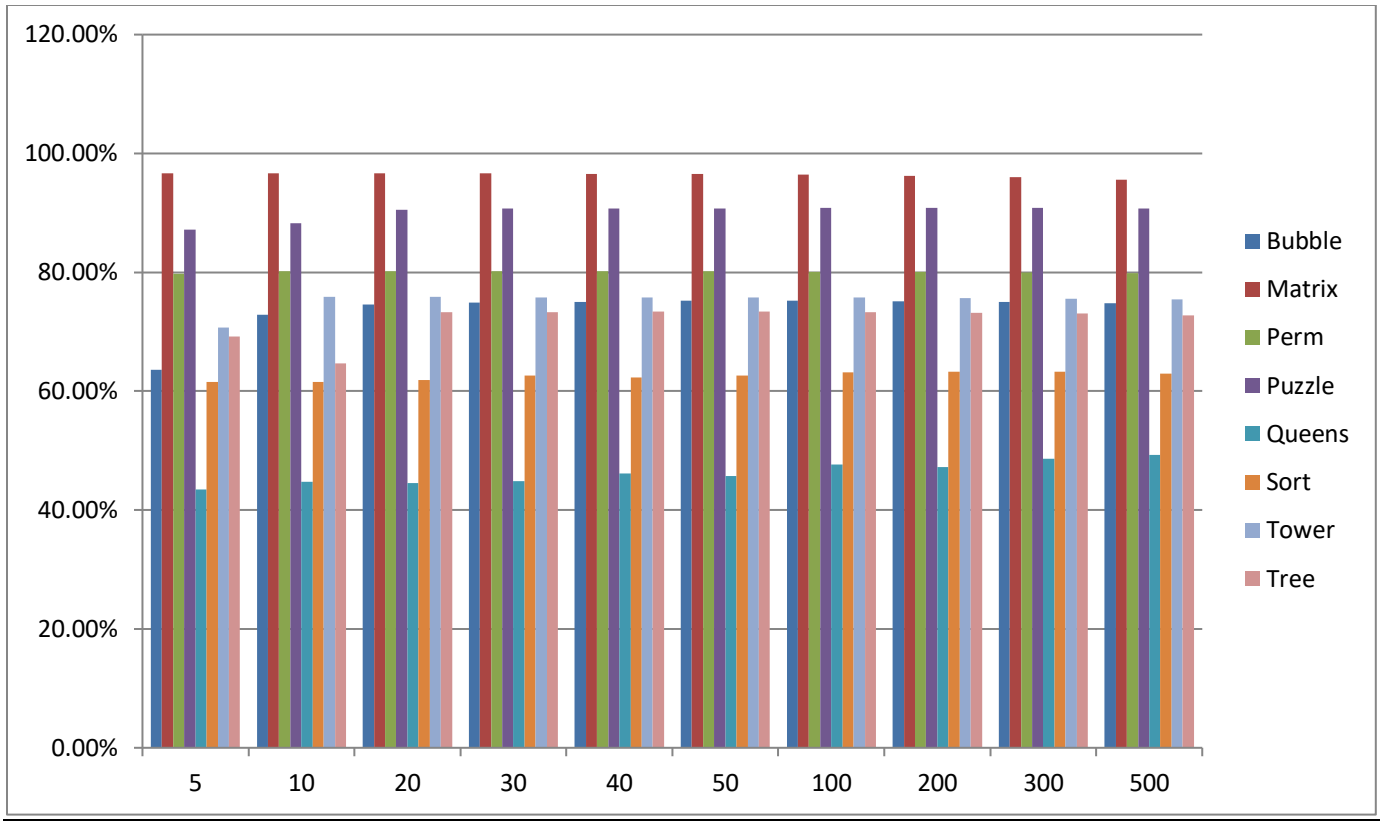
HRg	bubble	matrix	perm	puzzle	queens	sort	tower	tree	average
5	74.44%	93.39%	61.32%	86.31%	57.74%	61.27%	67.48%	59.06%	70.13
10	79.16%	96.33%	68.30%	88.05%	62.03%	61.87%	64.59%	74.39%	74.34
20	81.40%	96.56%	70.73%	89.59%	66.35%	63.58%	59.51%	73.27%	75.12
30	82.56%	96.56%	78.38%	89.67%	64.66%	63.36%	66.88%	72.21%	76.79
40	83.04%	96.45%	78.68%	90.06%	65.45%	63.43%	72.89%	70.49%	77.56
50	82.93%	96.45%	79.38%	90.46%	65.01%	63.75%	71.34%	70.68%	77.50
100	82.21%	96.66%	80.17%	90.89%	66.10%	63.80%	74.74%	74.43%	78.63
200	83.21%	96.68%	80.17%	91.15%	68.04%	64.55%	75.85%	75.01%	79.33
300	83.66%	96.69%	80.17%	68.44%	64.52%	75.85%	75.17%	91.46%	79.50
500	83.69%	96.69%	80.17%	68.96%	64.55%	75.85%	75.32%	91.61%	79.61



Din acest exercitiu putem observa ca acuratetea de predictie este crescatoare de la HRg = 5 pana la HRg = 500. De asemenea, cele mai mari valori au fost obtinute pentru benchmark-ul FMatrix.

Exercitiul 2

HRg	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Average
5	63.55%	96.68%	79.70%	87.20%	43.51%	61.59%	70.68%	69.15%	71.51%
10	72.84%	96.67%	80.14%	88.31%	44.73%	61.60%	75.85%	64.64%	73.10%
20	74.60%	96.65%	80.16%	90.51%	44.49%	61.89%	75.84%	73.28%	74.68%
30	74.94%	96.63%	80.16%	90.70%	44.84%	62.58%	75.82%	73.34%	74.88%
40	74.99%	96.60%	80.16%	90.74%	46.19%	62.33%	75.81%	73.42%	75.03%
50	75.19%	96.58%	80.15%	90.77%	45.76%	62.62%	75.81%	73.40%	75.04%
100	75.24%	96.46%	80.12%	90.88%	47.67%	63.12%	75.76%	73.33%	75.32%
200	75.15%	96.23%	80.05%	90.86%	47.23%	63.29%	75.69%	73.18%	75.21%
300	75.03%	96%	79.98%	90.84%	48.60%	63.33%	75.58%	73.03%	75.30%
500	74.79%	95.54%	79.84%	90.79%	49.32%	62.99%	75.40%	72.73%	75.18%



Din acest exercitiu putem observa ca mediile aritmetice pentru acuratetea de predictie sunt apropiate de la HRg = 5 pana la HRg = 500, deci variatia acestuia nu cauzeaza schimbari semnificative.

Laborator 9

Cerinte:

1. Care este numărul optim de noduri de pe nivelul intermediar în funcţie de istoria globală KG{2,4,6,8,10} a saltului pentru fiecare din paşii de învăţare stabiliţi ({0.125, 0.5, 1.00}) şi KL=0. Funcţia de activare utilizată în cadrul simulatorului este f(x)= 1 / (1 + e^{-x}) Concluzionaţi.
2. Cu numărul de noduri de pe nivelul ascuns stabilit determinaţi pasul optim de învăţare în funcţie de istoria globală a saltului.
3. Realizaţi comparativ graficul acurateţii de predicţie în condiţiile ne-antrenări reţelei şi antrenări clasice (cu preînvăţare pe baza analizei statistice, urmată de predicţie).
4. În condiţiile determinării numărului optim de noduri de pe nivelul ascuns (N2=N1+2) determinaţi procentul optim de filtrare a statisticilor, folosit în cadrul metodei de antrenare clasică a reţelei neuronale în funcţie de istoria globală (KG{2,4,6,8,10}) a saltului. Pragul de filtrare poate lua valorile: 60%, 70%, 80%, 90%, 95%.

Exercitiul 1

	alfa=0.5									
hidden Layer	HRG	fbubble	fmatrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	Avg
	2	85.47%	96.71%	88.61%	95.23%	79.64%	76.35%	96.97%	89.35%	88.54%
	4	85.43%	96.71%	92.85%	95.60%	81.55%	76.23%	97.01%	89.76%	89.39%
15	6	86.15%	96.71%	95.06%	95.49%	81.27%	74.73%	96.84%	89.52%	89.47%
	8	86.25%	96.71%	94.90%	95.39%	81.56%	-	-	-	90.96%
	10	86.16%	96.71%	94.62%	95.46%	83.06%	-	-	-	91.20%
	2	85.46%	96.70%	89.44%	95.32%	80.00%	76.46%	96.78%	89.48%	88.71%
	4	85.63%	96.70%	93.14%	95.69%	81.61%	76.18%	96.89%	89.73%	89.45%
30	6	85.71%	96.70%	95.04%	95.79%	82.59%	76.11%	96.88%	89.70%	89.82%
	8	86.32%	96.70%	94.23%	95.85%	82.80%	77.19%	96.60%	98.90%	91.07%
	10	86.21%	96.70%	94.28%	95.86%	83.90%	-	-	-	91.39%
	2	85.65%	96.69%	88.65%	95.26%	79.93%	75.84%	96.76%	89.40%	88.52%
	4	85.50%	96.70%	92.35%	95.63	82.02%	76.18%	97.03%	89.75%	89.40%
50	6	85.69%	96.70%	95.02%	95.75%	82.58%	76.81%	96.82%	89.70%	89.88%
	8	86.41%	96.70%	94.35%	95.85%	82.43%	75.42%	96.62%	90.02%	89.73%
	10	86.41%	96.70%	94.16%	95.86%	82.91%	-	-	-	91.21%

Din acest exercitiu putem observa ca cea mai buna medie a acuratetei de predictie se obtine pentru Hidden layer = 30, cu HRg = 8. (Excluzand

cazurile in care simularile nu au afisat niciun rezultat datorita timpului de simulare indelungat)

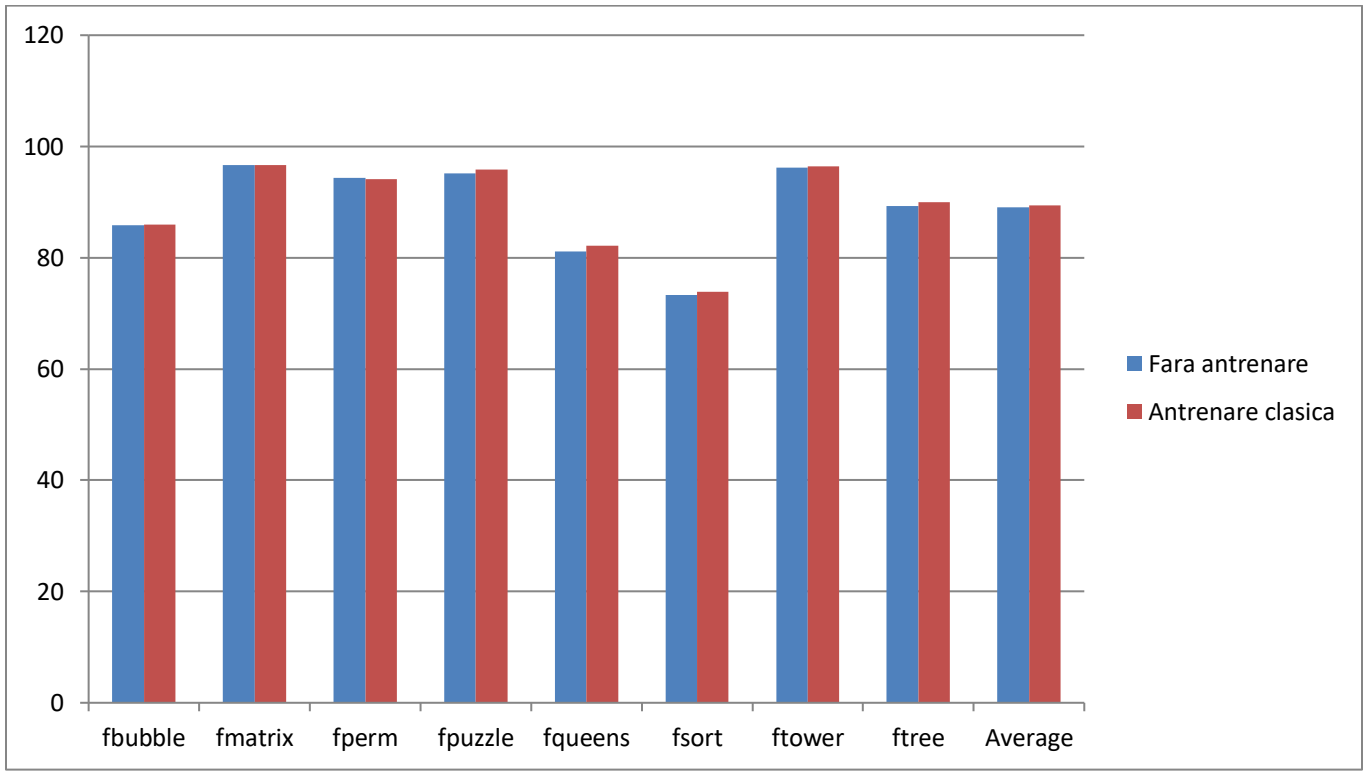
Exercitiul 2

Hidden layer	15				Acuratetea					
HRG	Learning step	fbubble	fmatrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	Avg
0	0,125	85,23	96,71	90,25	94,04	80,37	73,21	96,55	89,56	88,24
1	0,25	85,39	96,71	89,91	95,22	79,94	72,53	96,74	89,59	88,25375
2	0,375	85,68	96,71	88,43	95,29	80,45	76,34	96,8	89,7	88,675
3	0,5	86,03	96,71	89,11	95,47	81,73	77,55	97,03	89,74	89,17125
4	0,625	85,88	96,71	92,96	95,7	81,42	76,44	97,05	89,91	89,50875
5	0,75	85,73	96,71	93,28	95,76	81,37	77,5	97,01	89,71	89,63375
6	0,875	86,27	96,71	95,36	95,71	81,33	75,34	95,82	89,44	89,4975
7	1	86,11	96,71	94,86	95,4	81,62	75,88	96,63	89,78	89,62375
8	1,125	86,23	96,71	94,26	95,63	82,14	77,11	96,74	90,1	89,865
9	1,25	86,25	96,71	94,12	95,74	82,86		96,6	90,16	91,77714286
10	1,375	86,58	96,71	94,83	95,18	81,92		96,7	89,91	91,69

Din acest exercitiu putem observa ca cea mai buna medie a acuratetei de predictie se obtine pentru HRg = 8, cu Learning step = 1,125.
(Excluzand cazurile fara niciun rezultat afisat ca la primul exercitiu)

Exercitiul 3

Hidden Layer de la ex 1									
Learning Step de la ex 2									
					Acuratetea				
	fbubble	fmatrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	Average
Fara antrenare	85.84	96.7	94.33	95.2	81.14	73.35	96.24	89.35	89.01875
Antrenare clasica	85.94	96.7	94.1	95.9	82.14	73.93	96.44	89.95	89.3875



Din acest exercitiu putem observa ca diferenta dintre acuratetea de predictie in conditiile neatrenari retelei si a antrenari clasice este foarte mica.

Exercitiul 4

HRg	Filter	fbubble	fmatrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	Average
2	60,00%	85,74%	96,71%	89,37%	95,40%	80,01%	75,71%	97,01%	89,50%	88,68%
	70,00%	85,74%	96,71%	88,44%	95,38%	80,05%	76,79%	97,01%	89,42%	88,69%
	80,00%	85,74%	96,71%	88,10%	95,18%	79,14%	75,40%	97,01%	89,42%	88,34%
	90,00%	85,74%	96,71%	88,10%	95,25%	79,97%	75,75%	97,01%	89,34%	88,48%
	95,00%	85,29%	96,71%	87,55%	95,29%	79,82%	74,89%	96,74%	89,40%	88,21%
4	60,00%	85,20%	96,71%	93,36%	95,68%	81,48%	76,23%	97,03%	89,80%	89,44%
	70,00%	85,45%	96,71%	93,15%	95,63%	81,49%	76,84%	97,03%	89,79%	89,51%
	80,00%	84,94%	96,71%	93,26%	95,70%	81,44%	75,49%	97,03%	89,76%	89,29%
	90,00%	84,94%	96,71%	93,27%	95,65%	81,41%	75,35%	97,03%	89,66%	89,25%
	95,00%	84,94%	96,71%	93,27%	95,57%	81,37%	74,66%	97,06%	89,66%	89,16%
6	60,00%									
	70,00%									
	80,00%	85,63%	96,71%	94,77%	95,61%	82,10%	76,38%	96,87%	89,87%	89,74%
	90,00%	85,66%	96,71%	94,72%	95,68%	82,38%	76,62%	96,87%	89,69%	89,79%
	95,00%	85,56%	96,71%	94,72%	95,76%	82,25%	75,98%	96,54%	89,90%	89,68%
8	60,00%	86,11%	96,71%	95,20%	95,69%					93,43%
	70,00%	85,57%	96,71%	95,24%	95,83%	82,43%	74,36%	96,73%	89,33%	89,53%
	80,00%	85,89%	96,71%	94,80%	95,91%	82,26%	76,38%	96,60%	89,98%	89,82%
	90,00%	85,59%	96,71%	95,34%	95,41%	81,16%	74,03%	96,72%	88,54%	89,19%
	95,00%	85,58%	96,71%	95,41%	95,55%	80,90%	73,64%	96,73%	89,12%	89,21%
10	60,00%	86,42%	96,72%	95,38%	95,80%	82,72%				91,41%
	70,00%	86,36%	96,71%	95,07%	95,98%	83,42%				91,51%
	80,00%	86,58%	96,71%	95,76%	95,38%	93,31%				93,55%
	90,00%	85,43%	96,71%	94,76%						92,30%
	95,00%									

Din acest exercitiu putem observa ca, excluzand simularile care nu au afisat niciun rezultat, cele mai bune rezultate s-au obtinut pentru HRg = 8.

Laborator 10

Cerinte:

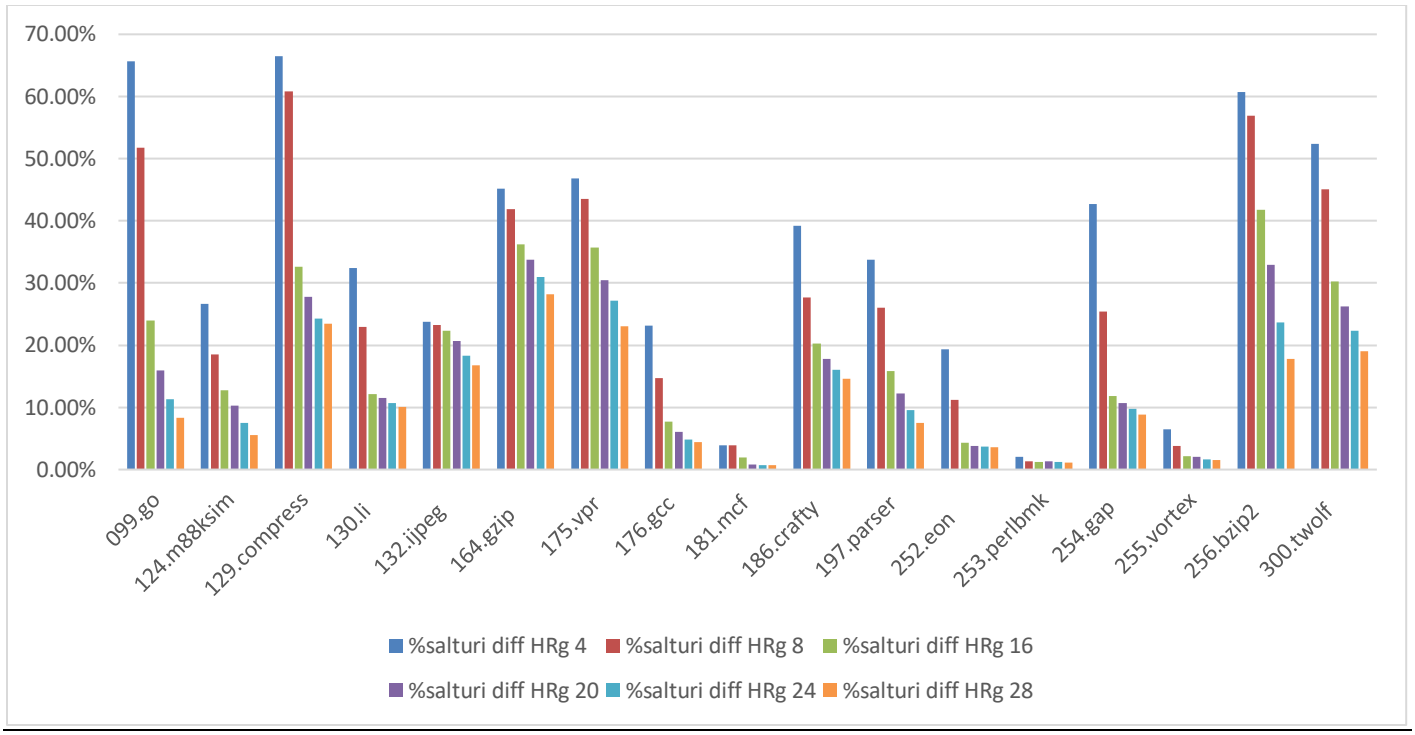
Sa se determine procentajul de salturi dificil de prezis pentru benchmark-urile SPEC2000 in urmatoarele situatii:

a) Fara istorie locala; Cu istorie globala HRg = {4, 8, 16, 20, 24, 28}
b) b) Cu istorie locala (alegeti o valoare); Cu istorie globala HRg = {4, 8, 16, 20, 24, 28}
c) Realizati o comparatie intre cele 2 grafice.

Exercitiul 1

a)

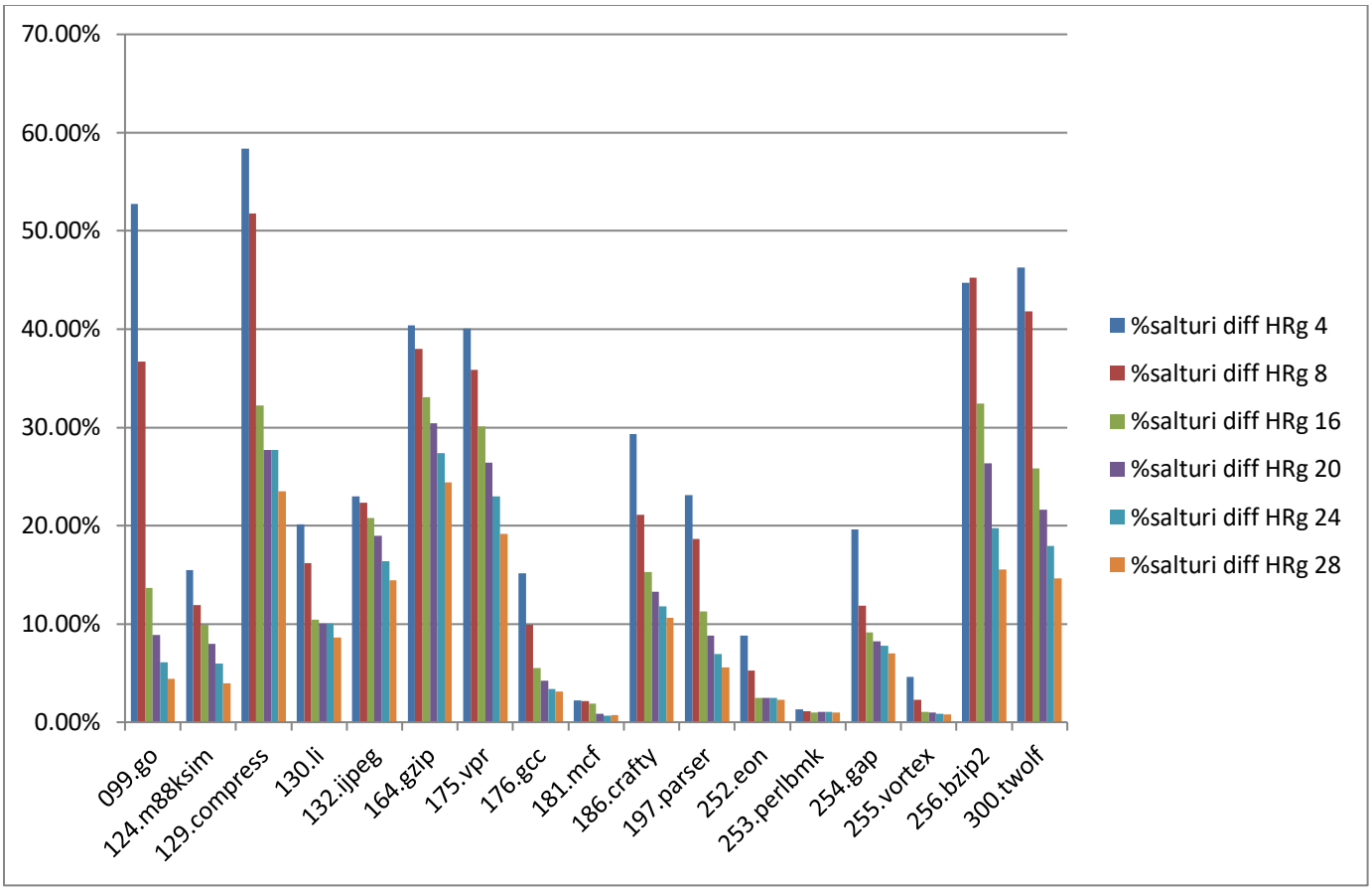
	099.go	124.m88ksim	129.compress	130.li	132.ijpeg	164.gzip	175.vpr	176.gcc	181.mcf	186.crafty	197.parse	252.eon	253.perlbm	254.gap	255.vorte	256.bzip2	300.twolf
%salturi diff HRg 4	65.69%	26.62%	66.50%	32.46%	23.79%	45.14%	46.83%	23.10%	3.87%	39.25%	33.73%	19.33%	2.05%	42.74%	6.44%	60.68%	52.34%
%salturi diff HRg 8	51.78%	18.48%	60.79%	22.95%	23.22%	41.88%	43.50%	14.67%	3.86%	27.67%	26.04%	11.24%	1.36%	25.41%	3.84%	56.95%	45.06%
%salturi diff HRg 16	23.97%	12.71%	32.63%	12.17%	22.33%	36.24%	35.66%	7.73%	1.90%	20.30%	15.80%	4.36%	1.23%	11.78%	2.19%	41.80%	30.20%
%salturi diff HRg 20	15.93%	10.33%	27.73%	11.50%	20.68%	33.80%	30.50%	6.06%	0.85%	17.79%	12.27%	3.77%	1.28%	10.66%	2.02%	32.91%	26.24%
%salturi diff HRg 24	11.30%	7.55%	24.31%	10.73%	18.29%	31.02%	27.13%	4.79%	0.70%	16.06%	9.57%	3.67%	1.20%	9.81%	1.64%	23.65%	22.33%
%salturi diff HRg 28	8.32%	5.56%	23.49%	10.06%	16.76%	28.17%	23.07%	4.44%	0.74%	14.60%	7.52%	3.56%	1.15%	8.82%	1.54%	17.79%	19.00%



Din acest subpunct putem observa ca procentul salturilor dificil de prezis fara istorie locala este din ce in ce mai mic cu cat marim dimensiunea istoriei globale de la HRg = 4 pana la HRg = 28.

b)

	099.go	124.m88ksim	129.compress	130.li	132.ijpeg	164.gzip	175.vpr	176.gcc	181.mcf	186.crafty	197.parser	252.eon	253.perlbmk	254.gap	255.vortex	256.bzip2	300.twolf
%salturi diff HRg 4	52.78%	15.50%	58.37%	20.16%	23.02%	40.39%	40.05%	15.14%	2.20%	29.34%	23.14%	8.83%	1.28%	19.63%	4.59%	44.74%	46.24%
%salturi diff HRg 8	36.68%	11.92%	51.77%	16.18%	22.32%	38.01%	35.85%	9.91%	2.14%	21.13%	18.62%	5.24%	1.09%	11.88%	2.25%	45.25%	41.80%
%salturi diff HRg 16	13.67%	9.93%	32.25%	10.43%	20.81%	33.11%	30.11%	5.54%	1.90%	15.27%	11.30%	2.50%	0.99%	9.11%	1.06%	32.44%	25.81%
%salturi diff HRg 20	8.89%	7.95%	27.73%	10.03%	19.00%	30.44%	26.41%	4.25%	0.83%	13.29%	8.82%	2.49%	1.06%	8.22%	0.97%	26.34%	21.65%
%salturi diff HRg 24	6.13%	5.96%	27.73%	10.03%	16.39%	27.40%	22.98%	3.37%	0.69%	11.81%	6.96%	2.46%	1.02%	7.76%	0.84%	19.73%	17.94%
%salturi diff HRg 28	4.40%	3.97%	23.49%	8.61%	14.47%	24.38%	19.19%	3.15%	0.72%	10.63%	5.61%	2.29%	1.02%	7.03%	0.82%	15.54%	14.66%



Din acest subpunct putem observa ca procentul salturilor dificil de prezis cu istorie locala = 4 este din ce in ce mai mic cu cat marim dimensiunea istoriei globale, ca la subpunctul a).

c)

Cand simularile sunt facute fara istorie locala, valorile sunt mai mari fata de subpunctul b), unde simularile au fost facute cu istorie locala = 4. Asadar, istoria locala ajuta la predictionarea salturilor greu de prezis.