Name: Lawrence Bush
RCS Account: bushl2

Section: 1
Email: Lawrence_Bush@dps.state.ny.us

Project 4

TABLE OF CONTENTS

INTRODUCTION

The program (NBA) implements a restricted backtracking algorithm to select 10-12 players for a basket ballteam. The goal is to mazimize the sum of the players ratings.

The team must also satisfy the following constraints:

1. A team can not have less than 10 players.
2. A team can not have more than 12 players.
3. A team must have at least 4 guards.
4. A team must have at least 3 forwards.
5. A team must have at least 1 center.
6. Total sum of players salaries must be less than or equal to the salary cap.

MAKEFILE: NBA.mak

PURPOSE OF EACH SOURCEFILE:

NBA.cpp - Main program.
Reads in player data file.
Creates draft list of players.
Implements restricted backtracking algorithm.

player.h - Contains a player class.
The player object stores player information, and performs formated output as well as other functions on player data.

player.cpp - Contains functions for the player class.

draft_list.h - Contains a draft_list class.
Stores a vector of players read in from file.
Performs functions on the vector such as

sort and accessor [].

draft_list.cpp -Contains functions for the draft_list class.

team.h - Contains a team class.
Stores a vector of players (team) that are inserted (hire) from the draft_list.
Performs various functions on the team such as hire, cut, criteria (does team meet the criteria), and print.

team.cpp - Contains functions for the team class.

COMPILE AND RUN INSTRUCTIONS:

Unzip all files into a folder.
Open NBA.cpp in MS Visual C++.
Build a project.
Add the files player.cpp, draft_list.cpp, team.cpp to the project.
Compile and run(! button).

The .h files (player.h, draft_list.h, team.h) should be in the same directory as NBA.cpp, as well as the .cpp function files.

Command line arguments:

Note: If you input the command line arguments incorrectly, the program will instruct you on how to input them.

The NBA program takes as command line arguments;
two file names (input file of players and an output file),
a salary cap, a sorting criteria and 12 numbers which
determine the scope (number of nodes for each level) of the
tree exploration.

example:

players.txt output.txt 30000000 salary 50 20 10 5 2 1 1 2 5 10 20 50

RESULTS -

I tested the program using the following configurations:

Normal Scope
50 10 5 2 1 1 1 1 1 2 5 10

		Criteria		
Cap	Salary	Rating	Value	C1
30M	0	234	221	248
35M	0	246	221	269
40M	0	259	221	295
Average	0	246	221	271

Increased Depth Scope

100 5 2 1 1 1 1 1 1 1 2 50

Criteria

Cap	Salary	Rating	Value	C1
30M	0	200	209	241
35M	0	221	209	250
40M	183	247	209	288
Average	61	223	209	260

Increased Breadth Scope
3 3 3 3 3 3 3 3 3 3 3

Criteria

Cap	Salary	Rating	Value	C1
30M	0	215	137	239
35M	0	270	137	263
40M	0	277	137	294
Average	0	254	137	265

Scope:

The experiment tested three scope configurations,
four criteria, and three salary caps for a total of 36 configurations.

The three scope configurations that I explored a tree with
I refer to as normal, increased depth and increased breadth.
The configurations test about the same number of teams and
have similar running times. The normal is a configuration which
works well in many situations(works best on the average).
Increased depth goes deeper, by having a larger number in the
begining and end, but lower numbers is thin in the middle.
The increase breadth does not go very deep but explores more
combinations of the players at the beginning of the vector.

The experiment shows that the deeper exploration works well for
the salary criteria and the increased depth scope works well for the
rating and C1 criteria but poorly for the value criteria.

Criteria:

The four criteria I tested are Salary, Rating, Value and C1.

Salary is a pretty good ranking system (it puts them in
a useful order) but you have to go too deep to get to the
desirable players. In order for it to be useful, you have to
go very deep and wide which takes a very long time.

Rating works better. It puts the good players at the top
without regard to their salary. With a little more depth and breadth,
a very good team can be found.

Value does not work well for the salary caps tested.
You have to go very deep to get to the highly rated players.
The highly rated players are not grouped together either.
It would work better for a very low salary cap.

C1 is a criteria I developed.
It multiplies rating by value, with value curved by 45
(rating * (value + 45)). The curve (45) decreases the variation in value and consequently puts more emphasis on rating. It still, however, puts significant emphasis on value. C1 also gives centers additional points proportional to their value. This makes the algorithm more likely to include a center in the team, and especially those centers who have a respectable value. The C1 criteria also does an additional sort by value on the top 1/9 of the draft list which puts the highest value players of that part, at the top. It then sorts the top 1/16 by rating which puts these high value high rating players in order by rating. It then sorts the last 7/8 of the list by rating which allows the players in the lower part of the list to be selected in order of rating.

C1 works better than all of the other criteria.
Rating, which is the second best criteria, worked better than C1 in one configuration out of 9.

RESULTS -

The following are the best teams I discovered along with the configuration I used to explore the list:

Salary Cap: $ 30,000,000
Criteria : C1
Levels: 4 2 2 1 1 1 1 5 5 20 20 10

Print Team

Name	Pos.	Height	Weight	Points	Rebounds	Assists	Salary	Rating
Duncan,Tim	F	7ft 0in	248	23.2	12.4	3.2	3858240	32.15
Cassell,Sam	G	6ft 3in	185	18.6	3.7	9.0	3500000	28.49
Carter,Vince	G	6ft 7in	215	25.7	5.8	3.9	2267280	28.21
Jones,Eddie	G	6ft 6in	200	20.1	4.8	4.2	2500000	26.59
Bibby,Mike	G	6ft 2in	190	14.5	3.7	8.1	3092400	24.77
Brand,Elton	F	6ft 8in	260	20.1	10.0	1.9	3375960	24.14
Francis,Steve	G	6ft 3in	194	18.0	5.3	6.6	3020520	24.13
Pierce,Paul	F	6ft 7in	220	19.5	5.4	3.0	1503960	22.64
McGrady,Tracy	G	6ft 8in	210	15.4	6.3	3.3	1767120	22.23
Odom,Lamar	F	6ft 10in	220	16.6	7.8	4.2	2445480	21.42
Nowitzki,Dirk	F	6ft 11in	237	17.5	6.5	2.5	1583040	20.69
Miller,Oliver	C	6ft 9in	325	6.3	5.1	1.3	510000	13.22

Team Rating : 288.67
Team Salary : 29424000Press any key to continue

Salary Cap: $ 35,000,000
Criteria : C1
Levels: 1 2 1 1 1 1 1 5 5 20 50 10

Print Team

Name	Pos.	Height	Weight	Points	Rebounds	Assists	Salary	Rating
Duncan,Tim	F	7ft 0in	248	23.2	12.4	3.2	3858240	32.15
Cassell,Sam	G	6ft 3in	185	18.6	3.7	9.0	3500000	28.49
Carter,Vince	G	6ft 7in	215	25.7	5.8	3.9	2267280	28.21
Jones,Eddie	G	6ft 6in	200	20.1	4.8	4.2	2500000	26.59
Bibby,Mike	G	6ft 2in	190	14.5	3.7	8.1	3092400	24.77
Brand,Elton	F	6ft 8in	260	20.1	10.0	1.9	3375960	24.14
Francis,Steve	G	6ft 3in	194	18.0	5.3	6.6	3020520	24.13

readme

Pierce,Paul	F	6ft	7in	220	19.5	5.4	3.0	1503960	22.64
McGrady,Tracy	G	6ft	8in	210	15.4	6.3	3.3	1767120	22.23
Odom,Lamar	F	6ft	10in	220	16.6	7.8	4.2	2445480	21.42
Kidd,Jason	G	6ft	4in	212	14.3	7.2	10.1	6858335	32.19
Miller,Oliver	C	6ft	9in	325	6.3	5.1	1.3	510000	13.22

Team Rating : 300.19
Team Salary : 34699295

Salary Cap: $ 40,000,000
Criteria : C1
Levels: 1 1 1 1 1 1 1 2 2 20 10

Name	Pos.	Height		Weight	Points	Rebounds	Assists	Salary	Rating
Duncan,Tim	F	7ft	0in	248	23.2	12.4	3.2	3858240	32.15
Hill,Grant	F	6ft	8in	225	25.8	6.6	5.2	6939000	30.11
Cassell,Sam	G	6ft	3in	185	18.6	3.7	9.0	3500000	28.49
Carter,Vince	G	6ft	7in	215	25.7	5.8	3.9	2267280	28.21
Jones,Eddie	G	6ft	6in	200	20.1	4.8	4.2	2500000	26.59
Bibby,Mike	G	6ft	2in	190	14.5	3.7	8.1	3092400	24.77
Brand,Elton	F	6ft	8in	260	20.1	10.0	1.9	3375960	24.14
Francis,Steve	G	6ft	3in	194	18.0	5.3	6.6	3020520	24.13
Pierce,Paul	F	6ft	7in	220	19.5	5.4	3.0	1503960	22.64
McGrady,Tracy	G	6ft	8in	210	15.4	6.3	3.3	1767120	22.23
Kidd,Jason	G	6ft	4in	212	14.3	7.2	10.1	6858335	32.19
Cato,Kelvin	C	6ft	11in	255	8.7	6.0	0.4	1299000	13.87

Team Rating : 309.52
Team Salary : 39981815