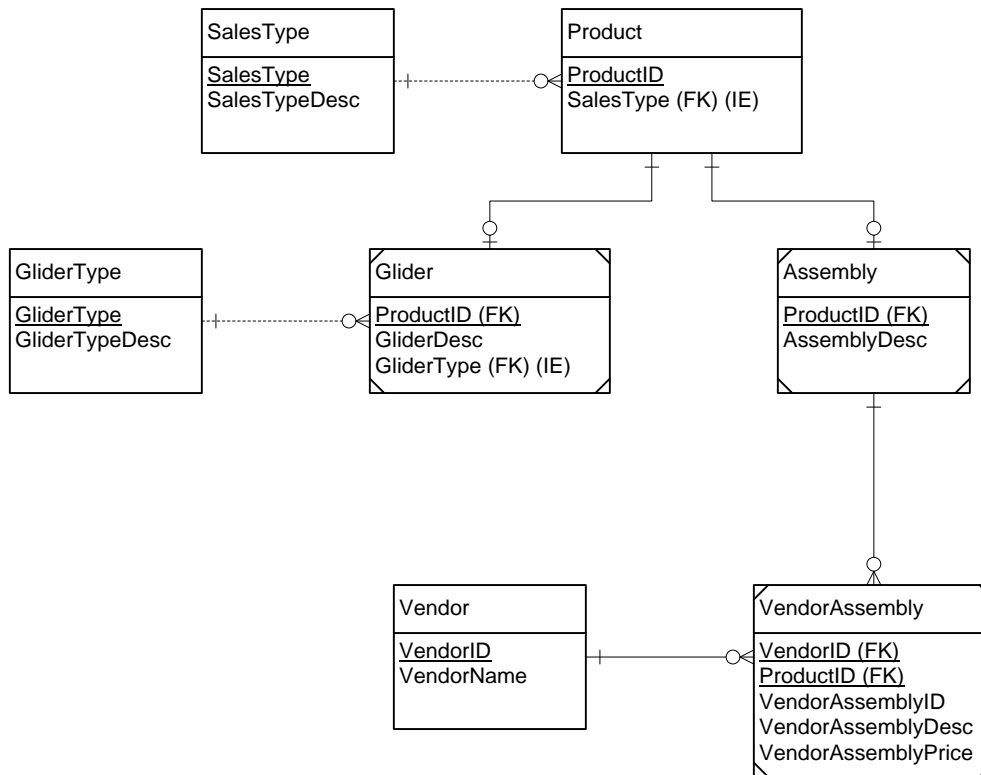


1.204 Quiz 1 Spring 2010 Solutions

1. Data model (35 points)



2. Database/SQL. (30 points)

Based on the data model in question 2, you construct a database where the tables, attributes and relationships correspond exactly to the data model structure. You are now asked to write the following SQL queries against your database.

- a. List the name and vendor ID of all vendors that supply at least one assembly to MITG. List each vendor only once. (10 points)

```
SELECT DISTINCT Vendor.VendorName, Vendor.VendorID FROM Vendor, VendorAssembly
WHERE Vendor.VendorID= VendorAssembly.VendorID;
```

- b. List the product ID and glider description of all high performance hang gliders with a wholesale sales type. Assume the code for high performance is "H" and for wholesale is "W" (10 points)

```
SELECT Glider.ProductID, Glider.GliderDesc FROM Product, Glider WHERE Product.ProductID=
Glider.ProductID AND Product.SalesType= "W" AND Glider.GliderType= "H";
```

- c. List the name and vendor ID of all vendors whose average assembly price is above \$50. (10 points)

```
SELECT Vendor.VendorName, Vendor.VendorID
FROM Vendor, VendorAssembly WHERE Vendor.VendorID = VendorAssembly.VendorID
GROUP BY Vendor.VendorName, Vendor.VendorID
HAVING (AVG(VendorAssembly.VendorAssemblyPrice)>50);
```

3. Algorithm design. (35 points)

Each job in the figure below starts at a time s_i and finishes at a time f_i . Two jobs are compatible if they do not overlap. Find the maximum subset of compatible jobs. Give an algorithm that solves this problem.

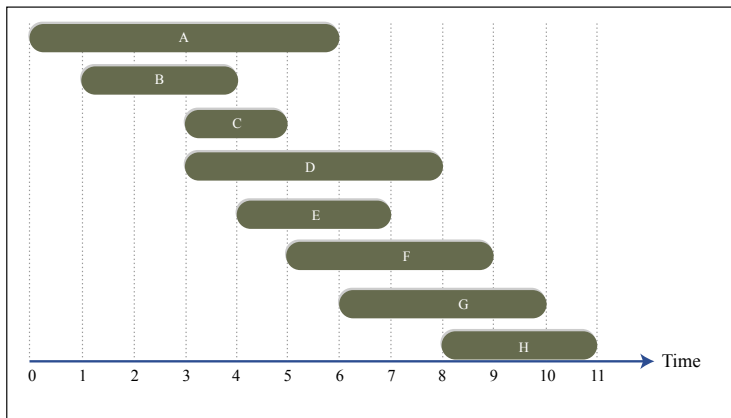


Figure by MIT OpenCourseWare.

- a. What kind of algorithm is it: divide and conquer, greedy, etc.?

Greedy.

- b. Write the algorithm in pseudocode or Java; comment or explain it so that it is easy to understand. **Sort jobs by earliest finish time. Try to add each job in order.**

Sort jobs by finish times so that $f_1 \leq f_2 \leq \dots \leq f_n$

Solution set A is empty at start

for $j = 1$ to n {

if (job j compatible with A) // Compare j to all jobs in A. If incompatible, break loop

Add j to A

}

return A

c. Informally show that your algorithm is correct.

Counterexamples for greedy algorithms that aren't quite right:

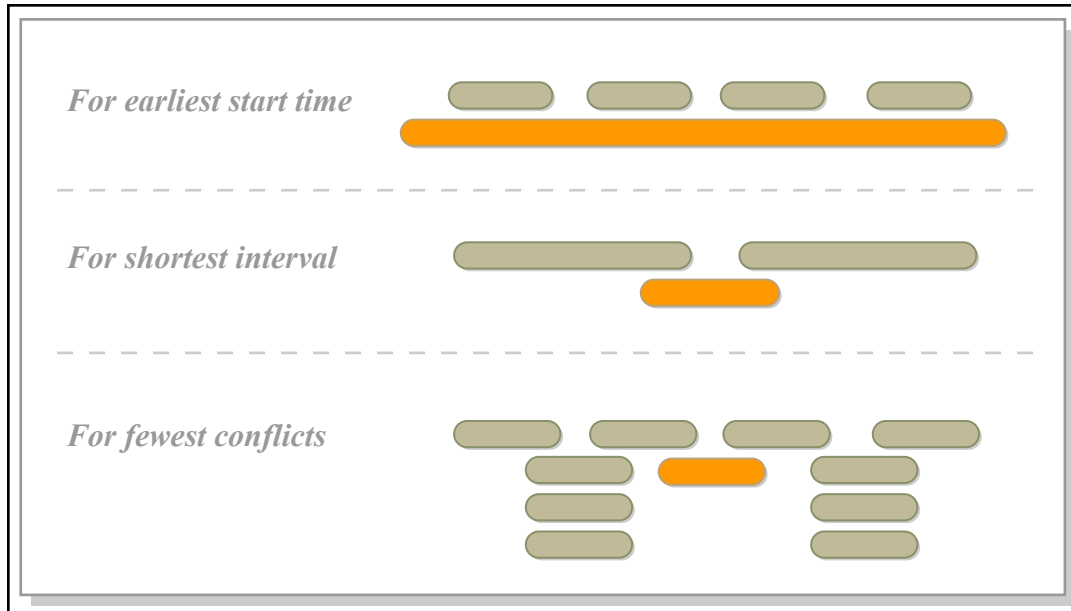


Figure by MIT OpenCourseWare.

Optimal solution: Sort jobs by earliest finish time. Proof by contradiction. Select initial set of jobs in greedy. Assume a job with a later finish time is in the optimal set. But this is impossible, since it uses more of the scarce time resource and gains nothing

MIT OpenCourseWare
<http://ocw.mit.edu>

1.204 Computer Algorithms in Systems Engineering
Spring 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.