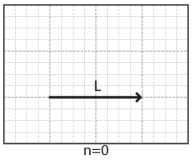
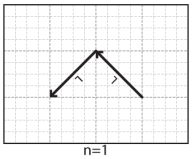
2010Fa CS10 Paper Final Answers

Question 1: Compare with empirical data. In the case of climate simulation, that means comparing against weather data from the past and seeing if the predicted weather is what actually happened (e.g., the 1938 hurricane that hit New York).

Question 2: It's an era of mobile, social and ubiquitous computing. Examples? Tweeting / facebook from a smartphone, exploring a city via foursquare, using Urbanspoon or Yelp to choose a restaurant, etc.

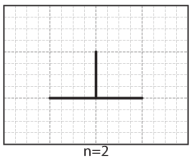
Question 3: They both must be functions, i.e., whose value is dependent only on the input and not on some prior state. In addition, the reducer should be associative and commutative. For the simplified model that we demonstrated in class, the domain of the reducer must include the range of the mapper AND the range of the reducer (not required for full credit).

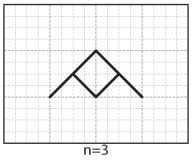




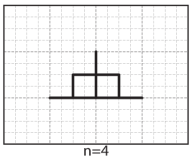
Question 4: "Concern about..."

1. service availability (trusting your business to someone else)
2. data lock-in (getting your data out when needed)
3. security / confidentiality / auditability
4. data transfer bottlenecks
5. performance unpredictability
6. scalable storage (as easy as computation)
7. bugs in large-scale distributed systems
8. the ability to scale up quickly
9. reputation fate sharing. E.g., spam-prevention, unexpected downtime for subpoenas, etc
10. software licensing (current non-open-source software pricing models don't scale well)



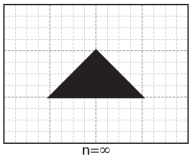


Question 5: Since chess can't be solved, chess AIs make moves based on "best guess" evaluations of the board. Dan's research group focuses on smaller games (and puzzles) that can be *strongly solved*, so his system *knows* the answer, once and for all, and plays perfectly.



○○○

Question 6: Playing chess, voice recognition, natural language translation, gesture recognition, driving trains, etc.



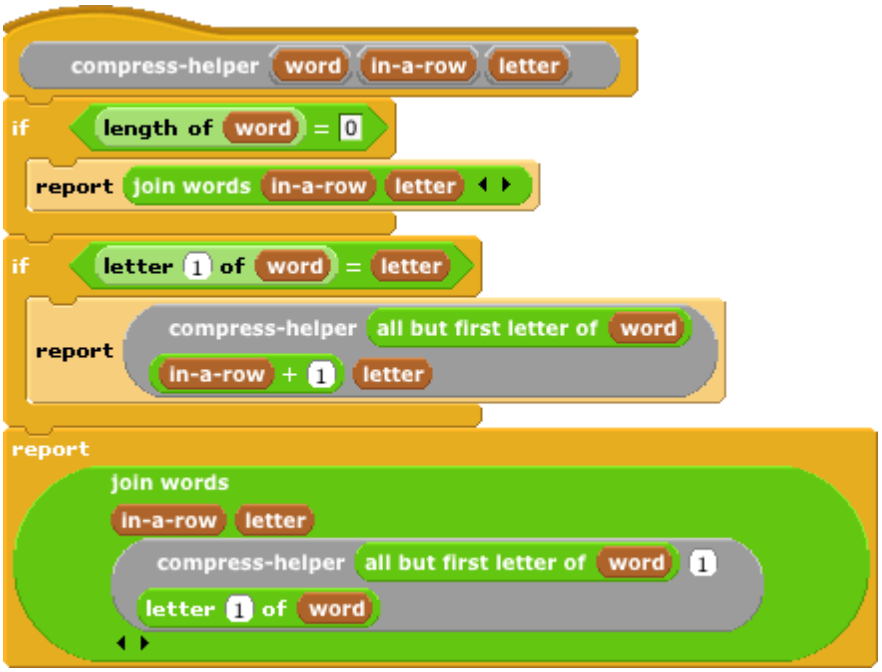
Question 7: (a) Halting problem. (b) exponential.

Question 8: My answer is "optimistic," although a case could be made for either. Some examples: They say in the last chapter that they think the Internet will eventually penetrate even currently isolated populations such as North Korea. They think that decreasing regulation will solve technological problems. They predict that people will come not to mind the

loss of their privacy (although they have mixed feelings about that themselves).

Question 9: (a) The n = 2, 3, 4, and ∞ generations for the fractal are above. (b) Exponential.

Question 10: (a) The first reported value changes to "`join-words(in-a-row, letter)`"
(b) `compress("hi")` returns "`1 h 1 h`" instead of "`1 h 1 i`"; the fix involves changing the last reported value to `compress-helper(all-but-first-letter-of(word), 1, letter(1)of(word))`
(c) 399. Alternating letters (e.g., "`hihi`...") double in size with spaces "`1 h 1 i 1 h 1 i`..."



Question 11: Change the value returned in the `report` block on line 6 to (4 options, depending on whether they used our MapReduce or not, and grey borders or `the( )block` green block):

`Map[ path-home? ]Reduce[ or ]over( go-neighbors(PLACE) )`
or
`Map[ path-home?(_) ]Reduce[ (_)or(_) ]over( go-neighbors(PLACE) )`

`Map(the(path-home?)block)Reduce(the(or)block)over( go-neighbors(PLACE) )`
or
`Map(the(path-home?(_))block)Reduce(the((_)or(_))block)over( go-neighbors(PLACE) )`


```
combine-with-[ or ]-items-of( map[ path-home? ]over( go-neighbors(PLACE) ) )
```

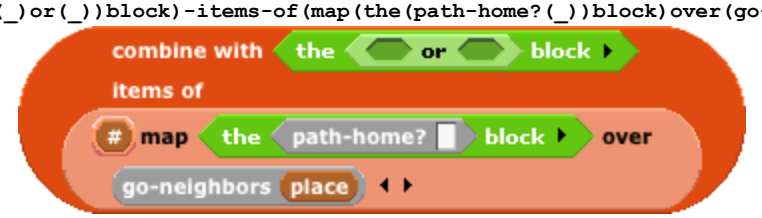
or

```
combine-with-[ (_)or(_) ]-items-of( map[ path-home?(_) ]over( go-neighbors(PLACE) ) )
```


```
combine-with-(the(or)block)-items-of( map(the(path-home?)block)over( go-neighbors(PLACE) ) )
```

or

```
combine-with-(the((_)or(_))block)-items-of(map(the(path-home?(_))block)over(go-neighbors(PLACE)))
```



Question 12:

a. 5. Assuming the biggest space that *cannot* hold a car is = 0.99, we place the cars from back to front so that we always leave a 0.99 spot between cars, the worst possible.
   1. 0.99 - 1.99
   2. 2.98 - 3.98
   3. 4.97 - 5.97
   4. 6.96 - 7.96
   5. 8.95 - 9.95

b. It's a simple divide-and-conquer recursion: