AL Beta Revision Report

Feedback comment:
“Overall, I think you’ve done a wonderful job. As someone who teaches both Data Structures (CS-2) and a Single CS-core-only Algorithms course, I think your topics and their placement are for the most part spot-on. I have a few comments, especially on sorting and search.”

How incorporated:
On behalf of the AL Sub-Committee, thanks.

Why not incorporated:

Date considered:

Feedback comment:
AL-Fundamentals WRT examples of sorting algorithms listed by worst case complexity:
“Lower bounds on the problem of comparison based sorting aren't in this list? Is that a deliberate omission?”

How incorporated:
Lower bounds are addressed in AL-Complexity “Best, average, and worst case performance of an algorithm”. The objective in AL-Fundamentals was to list three types of common sorting algorithms by worst-case complexity.

Why not incorporated:

Date considered:

Feedback comment:
AL-Fundamental: WRT Hamiltonian This is a concept or a graph property, not an algorithm. Are we discussing the concept within graph theory or an algorithm to find

How incorporated:
Moved to a possible example of Factorial complexity under AL-Complexity

Why not incorporated:
Survey feedback
Feedback comment:

AL-Strategies: WRT to list of Divide and Conquer algorithm examples list:

“This strikes me as a very short list of D&C algorithms; the integer multiplication algorithm would be a natural one to add here, as would be binary exponentiation / modular exponentiation.”

How incorporated:

Why not incorporated:

Based on feedback from the community, “they” do not want specific algorithm instances specified, but wish to be free to choose their own representative examples. The committee would like to list representative examples, but thought to keep the list short. The “e.g.,” identification subsequently adopted by the committee attempts to capture this trade off.

Feedback comment:

AL-Strategies WRT Greedy:

“I’m somewhat surprised Subset Sum and Knapsack weren’t on this list.”

“This strikes me as a very short list of greedy algorithms; earliest deadline first is a natural fit that follows elsewhere in the field, as is the optimality of farthest-in-future caching.”

How incorporated:

As in the previous comment, there is a tradeoff of enumerating the strategies vs. allowing the community to select. However, since Knapsack is referred to in various places, it was added here too.

Why not incorporated:

Feedback comment:

WRT Core core hours and complexity in a CS1 course:

“Insignificant amount of time on this topic? Or is that not including using it in conjunction with other topics?”
How incorporated:
The thought is that it is included in conjunction with other topics. However, the Committee is trying to not prescribe exactly how and where topics are addressed, but provide a representative example. There is an assumption that every program will address complexity in their CS2 type course and that most programs will have a dedicated Algorithms course addressing complexity. There are about the same core hours dedicated to complexity as in CS2013, for what it’s worth. Algorithms was consistently driven smaller by feedback from the community.

Why not incorporated:

Date considered:

Feedback comment:
WRT including SHA-256 in a CS2 course
“This strikes me as an oddly specific choice for CS2. Will students be able to understand the work and why? Or is it to be presented as a black box?”

How incorporated:
Ultimately this topic was moved to a non-core topic.

Why not incorporated:

Date considered:

Feedback comment:
WRT Cryptography
“Since most actual data is encrypted with symmetric ciphers such as AES, this should be included as well.”

How incorporated:
Based on feedback from the community Cryptography was moved to non-core.

Why not incorporated:

Date considered:
Feedback comment:

WRT Heuristic graph search

“In teaching a junior-level algorithms course, I have found that approaching search starting with the generalized idea of a frontier and seeing breadth-first, depth-first, Dijkstra, and A* search as specific examples of searches, leads to better understanding of search in general.”

How incorporated:

Why not incorporated:

This is a great comment, but it primarily refers to pedagogy, which isn’t really a focus in this curricular guidelines.

Date considered:

Feedback comment:

“I hesitate to think of Strassen’s algorithm as “core”, It’s a good example of divide-and-conquer, but the reasoning behind it doesn’t seem easy to apply to other problems. KA Core, sure, though.”

How incorporated:

Listed as one possible example of Divide and Conquer. As mentioned elsewhere in this Revision Report, the community desired to choose what approach they used, but the Committee felt it important to give some examples.

Why not incorporated:

Date considered:

Feedback comment:

“Everywhere else you refer to the Floyd-Warshall algorithm, yet here you refer to two separate algorithms. I’m aware that Floyd and Warshall (and Roy) came up with similar results independently, but I think that we should be consistent.”

How incorporated:

It is now consistent

Why not incorporated:
Date considered:

Feedback comment:
WRT Foundational Complexity classes
How about “Polynomial,” “Maybe even replace “Cubic” with “the more general “Polynomial””

How incorporated:
Accepted. with Cubic as an example

Why not incorporated:

Date considered:

Feedback comment:
WRT Traveling sales person and approximation.
“I don’t see how this relates to either of the KA Core topics listed. I think this illustrative outcome belongs under “AL/Fundamental Data Structures and Algorithms”

How incorporated:
Perhaps. However, Approximation was moved to KA course and Traveling Salesperson is now used as an illustrative example that might be used to demonstrate the Brute-Force paradigm/strategy. TSP is really a problem and not an algorithm. There is some thought by the committee that students should know “classic” problems, but it was removed as a topic and now appears in the examples of various strategies. Probably an issue since problems and approaches are still confounded. Though, everyone probably understands the general intent.

Why not incorporated:

Date considered:

Feedback comment:
WRT required KAs and Tasks
“This seems to have a large Software Engineering component as well, especially in software requirements. I think the question of the form “what are the important states of the patient that should be tracked as they progress through the system” is a classic software requirements problem.”
How incorporated:
This is true, the tasks are being generalized and outside of inclusion in this AL KA.

Why not incorporated:

Date considered:

Feedback comment:

WRT Divide and Conquer

“Merge sort is a better choice, I think because the algorithm is more easily seen to be $O(n \log n)$, and unlike Quicksort, Merge Sort is $O(n \log n)$ in the worst case.”

How incorporated:
Agreed and both are now listed as potential ways to introduce divide and conquer. As mentioned elsewhere with the “e.g.” identifier balancing specifying examples with community choice given the community feedback.

Why not incorporated:

Date considered:

Feedback comment:

CDER response with respect to parallel

How incorporated:

Why not incorporated:
Community feedback suggest keeping this in the PD KA.

Date considered:

Feedback comment:

WRT adding Digital fingerprint, Black chain, Bloom filter

How incorporated:
**Why not incorporated:**

When these were added at various times, the community feedback was to not include them (primarily form Liberal Arts focused colleges in U.S). Also, they should be addressed under the security KA.

**Date considered:**

---

**Feedback comment:**

“Very uneven level of individual items: very simple and advanced items intermixed - abstract data types (very simple) followed by SHA-256 and RSA (advanced), followed by adjacency matrix of a graph (very simple), then nontrivial graph algorithms (medium to advanced) followed by linked lists (very simple), string matching (advanced) followed by simple sorting methods (medium), . . .”

“The order of items (the previous objection continued): I would expect very simple item first (ADT, linked lists, graph representations), then medium level (sorting, some graph algorithms), difficult last (some graph algorithms, string matching and RSA),“

**How incorporated:**

As addressed elsewhere in this report SHA-256 and RSA are no-longer core topics. An ordering from simple to complex makes sense in a simple list. The presentation of the topics has been completely changed. There is some debate of simple to complex versus a simple lexicographical ordering.

**Why not incorporated:**

**Date considered:**
Feedback comment:

“Even though the Strassen’s matrix multiplication is an ingenious algorithm, its not very high practical value and relatively high logical complexity make its inclusion to the fundamental algorithm list problematic; but it could be a nice illustration of the divide-and-conquer strategy”

How incorporated:

Agreed. As addressed elsewhere in this report. It has been moved into a list of one example that demonstrates Divide and Conquer. Again, based on community feedback, allowing the individuals to select which examples they will use.

Why not incorporated:

Date considered:

Feedback comment:

“to be well explained, differential privacy is not a simple notion, and hence I would not put it into this list; moreover, why differential privacy is here: the notion is quite important, but there are many other topics or algorithms that are at least as important and not included here (e.g., FFT that is among elective algorithms)”

How incorporated:

Perhaps, the community feedback is against including FFT, differential privacy has been moved to AL-SEP. However, (author note) I would disagree with the general assessment. I’ve been able to teach a differential privacy example in class in fifteen minutes in which 100% of students subsequently assessed were able to describe it’s purpose and use it for another example. This is impossible with FFT. However, I agree with the importance of FFT, it simply that too many in the community do not agree with its inclusion.

Why not incorporated:

Date considered:
Feedback comment:

“invariants: this item is very important from the point of view of methodology of teaching/learning of algorithms (see below), but it definitely does not belong to this list if I understand the topic in the same way as the task force members.”

How incorporated:

Why not incorporated:

Agreed. However, there were several members of the Steering Committee and the general community who argued for this being CS core.

Date considered:

Feedback comment:

“I do not agree with terms like social, ethical, secure and fair algorithms. From this point of view, an algorithm is neutral, it is neither good or bad, and it simply executes the code. It is the impact of the use of algorithms that is good or bad, fair or unfair.”

How incorporated:

Why not incorporated:

The philosophy literature would suggest that algorithms (and other things) are not value-neutral and that students need to understand this. The real question, which the broader comments get out, to a degree, is that what is or isn’t CS begs the real question that we educate computer scientist.

Date considered:

Feedback comment:

WRT If I had written an algorithmic curriculum myself

How incorporated:

Why not incorporated:

Author note: In general, I agree with comments and suggestions. However, based on the experience of working on CS202X. The CS education community is not ready to accept many of these suggestions.
Hence, as with the reviewer, many of us are left to implement things we believe in within our own programs. I believe a broader approach to CS3033 is required.

**Date considered:**

Feedback comment:

In Math requirements

"MSF/Basics of Probability Sample spaces, events, axioms Conditional Probability, Bayes Theorem Discr...”

**How incorporated:**
The fine granularity was removed and MSF-Discrete is incorporated as a whole, which includes probability outcomes.

**Why not incorporated:**

**Date considered:**

Feedback comment:

*Classification/Prediction Algorithms (e.g. Decision Tree, Regression)*

**How incorporated:**

Incorporated (subsequently removed) along with other KA topics based on

**Why not incorporated:**

**Date considered:**

Feedback comment:

Survey Result WRT Cryptography inclusion 53.2% Yes, 36.2% No

**How incorporated:**

Mixed results: topics moved to Non core.
Why not incorporated:

Date considered:

Feedback comment:
Survey Result WRT Hamiltonian Circuit inclusion  53.2% Yes, 34% No

How incorporated:
Included as one possible example “e.g.” for demonstrating O(n!) complexity.

Why not incorporated:
But not removed entirely from CS core.

Date considered:

Feedback comment:
Survey Result WRT Spanning Tree inclusion  76.6% Yes, 17% No

How incorporated:
Specific approaches are listed as “e.g.” choice examples

Why not incorporated:
Strong support for inclusion. Can also be used to demonstrate SEP (ala ITiCSE working group examples with CSG-Ed i.e., connecting villages with well water minimal cost).

Date considered:

Feedback comment:
Survey Result WRT Transitive Closure Warshall Tree inclusion  38.3% Yes, 44.7% No

How incorporated:
Moved from CS core to KA core

Why not incorporated:

Date considered:
Feedback comment:
Survey Result WRT String inclusion 50% Yes, 35% No

How incorporated:
Specific comments suggest the issue is the inclusion of Efficient String matching algorithms vs. strings. The matching algorithms have been moved from CS core to the KA core

Why not incorporated:

Date considered:

Feedback comment:
Survey Result WRT Differential Privacy inclusion 30% Yes, 45% No

How incorporated:
Moved to AL-SEP unit.

Why not incorporated:

Date considered:

Feedback comment:
Survey Result WRT Strassen’s Algorithm inclusion 57.5% Yes, 35% No

How incorporated:
Listed as one example “e.g.” for explaining Divide and Conquer.

Why not incorporated:

Date considered:

Feedback comment:
Survey Result WRT Approximation Algorithm inclusion 50% Yes, 40% No
How incorporated:
Moved from CS to KA core

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Branch-and-Bound Algorithm inclusion 53.3% Yes, 33.3% No

How incorporated:
Listed as an “e.g.” example as one possible approach to handle exponential growth.

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Branch-and-Bound Algorithm inclusion 53.3% Yes, 33.3% No

How incorporated:

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Consensus Algorithm inclusion 40% Yes, 50% No

How incorporated:
Moved from CS core to Non-core

Why not incorporated:
Date considered:

Feedback comment:
Survey Result Heuristic A* inclusion 59.4% Yes, 37.5% No

How incorporated:
Moved to an “e.g.” as one example way to handle exponential growth

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Heuristic A* inclusion 34.4% Yes, 43.8% No

How incorporated:
Moved from CS core to non-core (note included in PD KA)

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Randomized/Stochastic inclusion 31.3% Yes, 46.9% No

How incorporated:
Moved from CS core to KA core.

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Church/Turing Thesis inclusion 64.5% Yes, 32.3% No

How incorporated:
Why not incorporated:
Perhaps the single most important concept/result in computer science.

Date considered:

Feedback comment:
Survey Result Context aware computing inclusion 50% Yes, 34.6% No

How incorporated:
Moved from CS core to KA core of AL-SEP

Why not incorporated:

Date considered:

Feedback comment:
Survey Result Differential Privacy inclusion 38.5% Yes, 50% No

How incorporated:
Moved to AL-SEP as a possible “e.g.” example of algorithmic fairness, which has 80% support

Why not incorporated:

Date considered: