***CSCE686 2019  
Homework 3***

***Due (4/17) “groups of two are permissible”***

1. *(30pts)* Use the given domain/algorithm domain (PD/AD) design process for a P-Time problem\* by constructing an algorithm from PD through Functional Algorithm **using design refinement**. (Follow example design as presented in class – see suggested “short and sweet” beginning design outline indicated attached).

*\* Do not use MST or SSP*

1. *(20pts)*Test and evaluate the AFIT *graphprogram\** (on the L: drive under the CSCE686/NPC Problems/Software directory) for MIS and Clique examples that are small, medium and large graph applications **(planar & non-planar).** Present search tree for medium application. Some graph examples are in the *Graph* directory. Use at least 10 graphs. Utilize appropriate reporting approaches (**Barr, M&F-Appendix B.2**).

***References:***

1. *Bron-Kerbosch Algorithm*, Wikipedia (Checkout Wiki References) <http://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm>
2. *BHOSLIB,* *DIMACS,*  MIS/Clique Benchmark sets, Bibliography <http://iridia.ulb.ac.be/~fmascia/maximum_clique/BHOSLIB-benchmark>
3. *Floudas and Pardalos*, (eds.) Encyclopedia of Optimization, p. 411, Kluwer, 2009
4. *Wikipedia,* Software Testing, <http://en.wikipedia.org/wiki/Software_testing>
5. *Barr, et al*, Guidelines for Designing and Reporting on Computational Experiments with heuristic Methods, 2001 *“Great Reference!!”*
6. <https://www.google.com/?gws_rd=ssl#q=comparison+of+CLIQUE+algorithms>

***Report outline suggestion for the application of the PD/AD design process per class discussion and notes:***

1. **Title, Introduction**   
   - purpose of design exercise
2. **Define, develop, and analyze** Problem Domain (PD) Specification  
   - PD requirements – English to symbolic to math/logic formal representation  
    -- input domain and condition; PD data structures(s)  
    -- objective function  
    -- output domain and conditions; PD data structures(s)  
   - class of problem including size of solution space (references)
3. **Select** a Search Algorithm Domain (AD) based upon problem class  
   - for a P-Time problem, (P-Time Complexity, Matroid Class) use DFS – Greedy   
   - use the associated CSCE686 search algorithm template  
   - address the different specific algorithms available to solve this problem (references: CLRS, TR, …)

- the specific functional or OO algorithm selected to design; i.e., transform to via PD/AD design process

1. **Evolve** a general Algorithm Design Specification via integration  
   - integrate problem domain with search algorithm template  
   - this is the high-level algorithm design specification  
    -- input domains/output domains and conditions  
    -- algorithmic search model as instantiated from search template:  
    a. set of candidates  
    b. next-state generator  
    c. selection function   
    b. feasibility function  
    c. solution function  
    d. objective –fitness function
2. **Expand** Algorithmic Design Specification into operational form (functional or object-oriented) “ **Refine with numerous design steps”**

“Refine algorithm design recursively to pseudo code level”   
- input domains/output domains and conditions expanded  
- search specification is further defined and developed:  
 a. set of candidates  
 b. next-state generator  
 c. selection function   
 b. feasibility function  
 c. solution function  
 d. objective –fitness function  
- usually new data structures are defined for more efficient design  
- may involve various design refinement stages (5a, 5b, ..)  
- add comments to each stage of refinement to indicate the current evolution of the design using the search model elements (set of candidates, …) with ability to directly **flow** back design to requirements  
- leads to low-level functional or OO algorithmic design of Pseudo code  
 - usually mapping to specified programming language level   
- determine algorithm **complexity** at functional or OO level

* functional or OO level design should reflect “good” ADT design   
   - axiom definition NOT required but should be addressed
* your design process should explicitly reflect **rational** for development and use of **CREATIVE** data structures and control structures per refinement stage (design process should go slow!)

1. I**mplementation and Test and Analysis (extra credit)**- **if** functional or OO level mapping to executable code exists, then   
    - determine objective of your experiments and run  
    -- analyze time execution as compared to expectation  
    -- Use Barr’s et al suggestions for reporting on computational exp
2. **Conclusion**, Comments, Recommendations  
   -- address utility of PD/AD design process (effectiveness and efficiency)  
    -- development of specific functional or OO code selected  
    -- discuss possible development of other algorithmic code   
   - discuss impact of PD complexity vs. AD complexity (different!)  
   - briefly discuss variations of the selected problem (references)  
   - other general references and specific application references (limit!)  
   - relate to other SPP algorithms  
   ….
3. **References** (indicate all references)

Note:

1. Improve on the given top-down design development for better understandability and the proper use of heuristics. More details on   
   PD to AD integration and explicit design steps to functional pseudo code..
2. Use diagrams, figures and tables for conciseness and effect.
3. Should use Word or Latex manuscript software allowing more formalized reporting: table of contents, list of figures, index, …
4. It is presumed that one is doing a bottom-up design since the SPP algorithm implementation code or pseudo code is given. Nevertheless, the HW requires a top-design that stands on it s own ( A new reader would after reading your top-down design, would know exactly all the explicit design decisions leading to the SPP pseudo code.).
5. It is suggested that one work with a fellow student on this design development for better understanding of the design process. (one report per group)

***“For Homework 3,*** ***please send electronic version of the design solution via email”***