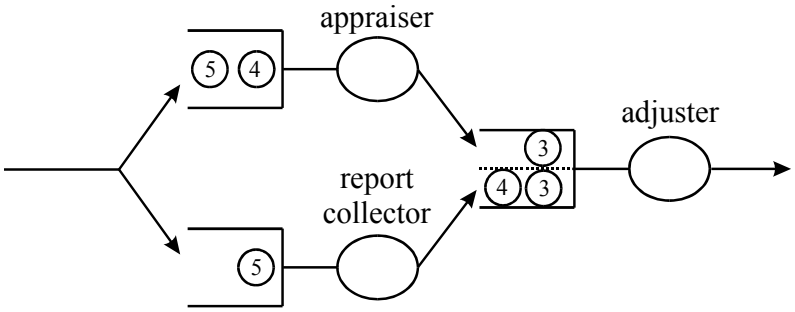
SAMPLE QUESTIONS: MIDTERM #1

1. (Claims processing) Consider a system for processing auto-accident insurance claims that consists of an appraiser, a report collector, and an adjuster. Insurance claims arrive to the system one at a time, with the interarrival times i.i.d. as a random variable A_1. When a claim arrives, one copy is sent to the appraiser, who prepares an estimate of the repair cost for the car and sends it to the adjuster. Simultaneously, another copy is sent to the report collector, who obtains an accident report from the police and sends it to the adjuster. The adjuster looks at the repair estimate and the accident report, and then sends an insurance payment to the claimant. Each person in the system processes claims one at a time, in the order that the claims arrive to the system. Note that the adjuster will not start to process a claim until both an accident report and a repair estimate become available. Nobody is idle when they can be working. The successive times required to prepare repair estimates (resp., to obtain accident reports) are i.i.d. as a random variable A_2 (resp., A_3). The successive times required by the adjuster to process claims are i.i.d. as a random variable A_4. All processing times are mutually independent, and all distributions are continuous, so that events never occur simultaneously. At time 0, a claim has just arrived to an empty system.

   Set $X(t) = (N_1(t), N_2(t), M(t))$, where $M(t)$ is the total number of claims in the system at time *t*, $N_1(t)$ is the number of claims in the system at time *t* that do not yet have a repair estimate and $N_2(t)$ is the number of claims in the system at time *t* that do not yet have an accident report. The figure below corresponds to the state $X(t) = (2,1,3)$: There are three claims in the system (numbered 3, 4, and 5), the appraiser is working on claim 4, the report collector is working on claim 5, and the adjuster is working on claim 3 (since both a repair estimate and accident report are available for this claim).



Define events as follows:

e_1 = "arrival of claim to the system"
e_2 = "completion of repair estimate for a claim"
e_3 = "completion of accident report for a claim"
e_4 = "completion of processing for a claim by adjuster"

a) The process $\{X(t): t \geq 0\}$ can be specified as a GSMP. Give the definition of (i) the state space S, (ii) the active event-set mapping $E(s)$, (iii) the speeds $r(s,e)$, and (iv) the state-transition probability function $p(s';s,e^*)$. **[You do not need to specify the clock-setting distribution functions or initial distribution.]**

b) Suppose that the first few new clock readings for events $e_1 - e_4$ are given in the following table:

C_1	C_2	C_3	C_4
4	2	3	4
5	6	4	1
7	3	8	2
5	2	6	4

Using these values, fill in the table below by doing a hand simulation of the GSMP from part (a). In the table, ζ_n denotes the time of the n^{th} state transition, $X(\zeta_n)$ the state just after the n^{th} state transition, $C_{n,i}$ the clock reading for event e_i just after the n^{th} state transition, and e^* the event that will trigger the next — i.e., the $(n+1)^{\text{st}}$ — state transition. (The first row has been filled in for you; the notation -- denotes an "empty" clock reading that corresponds to an event that is not currently active.)

n	ζ_n	$X(\zeta_n)$	$C_{n,1}$	$C_{n,2}$	$C_{n,3}$	$C_{n,4}$	e^*
0	0.0	(1,1,1)	4	2	3	--	e_2
1							
2							
3							
4							

c) Suppose that we have run our simulation five times and obtained the following five observations of D_{100}, the average time to process the first 100 claims: 1.0, 3.0, 7.0, 5.0, 4.0. Use the procedure given in class to calculate the approximate number of simulation runs needed to estimate $E[D_{100}]$ to within ±10% with 90% probability. (Use the same formula that you would use if the number of

pilot runs were much larger than 6, as would be the case in a real simulation.) **[You do not need to numerically compute the final answer; just set up the final computation.]**

2. Random number generation and parameter estimation. Consider a random variable X whose cdf is given by

$$F_X(x) = \begin{cases} 0 & \text{if } x < 0 \\ \beta x^{\alpha} & \text{if } 0 \le x \le \beta^{-1/\alpha} \\ 1 & \text{if } x > \beta^{-1/\alpha} \end{cases}$$

where $\alpha > 0$ and $\beta > 1$.

a) Give an algorithm for generating a sample of X that only requires a single uniform random number.

b) A friend gives you a copy of (the executable code of) their implementation of the algorithm from part (a). You run the algorithm, with $\alpha = 2$ and $\beta = 4$, a very large number of times (so that you have a very good estimate) and find that the average of the generated values is 0.16. Do you trust your friend's implementation? Explain why or why not.

TABLE T.1
Critical points $t_{\nu,\gamma}$ for the t distribution with ν df, and z_γ for the standard normal distribution
$\gamma = P(T_\nu \leq t_{\nu,\gamma})$, where T_ν is a random variable having the t distribution with ν df; the last row, where $\nu = \infty$, gives the normal critical points satisfying $\gamma = P(Z \leq z_\gamma)$, where Z is a standard normal random variable

	γ															
ν	0.6000	0.7000	0.8000	0.9000	0.9333	0.9500	0.9600	0.9667	0.9750	0.9800	0.9833	0.9875	0.9900	0.9917	0.9938	0.9950
1	0.325	0.727	1.376	3.078	4.702	6.314	7.916	9.524	12.706	15.895	19.043	25.452	31.821	38.342	51.334	63.657
2	0.289	0.617	1.061	1.886	2.456	2.920	3.320	3.679	4.303	4.849	5.334	6.205	6.965	7.665	8.897	9.925
3	0.277	0.584	0.978	1.638	2.045	2.353	2.605	2.823	3.182	3.482	3.738	4.177	4.541	4.864	5.408	5.841
4	0.271	0.569	0.941	1.533	1.879	2.132	2.333	2.502	2.776	2.999	3.184	3.495	3.747	3.966	4.325	4.604
5	0.267	0.559	0.920	1.476	1.790	2.015	2.191	2.337	2.571	2.757	2.910	3.163	3.365	3.538	3.818	4.032
6	0.265	0.553	0.906	1.440	1.735	1.943	2.104	2.237	2.447	2.612	2.748	2.969	3.143	3.291	3.528	3.707
7	0.263	0.549	0.896	1.415	1.698	1.895	2.046	2.170	2.365	2.517	2.640	2.841	2.998	3.130	3.341	3.499
8	0.262	0.546	0.889	1.397	1.670	1.860	2.004	2.122	2.306	2.449	2.565	2.752	2.896	3.018	3.211	3.355
9	0.261	0.543	0.883	1.383	1.650	1.833	1.973	2.086	2.262	2.398	2.508	2.685	2.821	2.936	3.116	3.250
10	0.260	0.542	0.879	1.372	1.634	1.812	1.948	2.058	2.228	2.359	2.465	2.634	2.764	2.872	3.043	3.169
11	0.260	0.540	0.876	1.363	1.621	1.796	1.928	2.036	2.201	2.328	2.430	2.593	2.718	2.822	2.985	3.106
12	0.259	0.539	0.873	1.356	1.610	1.782	1.912	2.017	2.179	2.303	2.402	2.560	2.681	2.782	2.939	3.055
13	0.259	0.538	0.870	1.350	1.601	1.771	1.899	2.002	2.160	2.282	2.379	2.533	2.650	2.748	2.900	3.012
14	0.258	0.537	0.868	1.345	1.593	1.761	1.887	1.989	2.145	2.264	2.359	2.510	2.624	2.720	2.868	2.977
15	0.258	0.536	0.866	1.341	1.587	1.753	1.878	1.978	2.131	2.249	2.342	2.490	2.602	2.696	2.841	2.947
16	0.258	0.535	0.865	1.337	1.581	1.746	1.869	1.968	2.120	2.235	2.327	2.473	2.583	2.675	2.817	2.921
17	0.257	0.534	0.863	1.333	1.576	1.740	1.862	1.960	2.110	2.224	2.315	2.458	2.567	2.657	2.796	2.898
18	0.257	0.534	0.862	1.330	1.572	1.734	1.855	1.953	2.101	2.214	2.303	2.445	2.552	2.641	2.778	2.878
19	0.257	0.533	0.861	1.328	1.568	1.729	1.850	1.946	2.093	2.205	2.293	2.433	2.539	2.627	2.762	2.861
20	0.257	0.533	0.860	1.325	1.564	1.725	1.844	1.940	2.086	2.197	2.285	2.423	2.528	2.614	2.748	2.845
21	0.257	0.532	0.859	1.323	1.561	1.721	1.840	1.935	2.080	2.189	2.277	2.414	2.518	2.603	2.735	2.831
22	0.256	0.532	0.858	1.321	1.558	1.717	1.835	1.930	2.074	2.183	2.269	2.405	2.508	2.593	2.724	2.819
23	0.256	0.532	0.858	1.319	1.556	1.714	1.832	1.926	2.069	2.177	2.263	2.398	2.500	2.584	2.713	2.807
24	0.256	0.531	0.857	1.318	1.553	1.711	1.828	1.922	2.064	2.172	2.257	2.391	2.492	2.575	2.704	2.797
25	0.256	0.531	0.856	1.316	1.551	1.708	1.825	1.918	2.060	2.167	2.251	2.385	2.485	2.568	2.695	2.787
26	0.256	0.531	0.856	1.315	1.549	1.706	1.822	1.915	2.056	2.162	2.246	2.379	2.479	2.561	2.687	2.779
27	0.256	0.531	0.855	1.314	1.547	1.703	1.819	1.912	2.052	2.158	2.242	2.373	2.473	2.554	2.680	2.771
28	0.256	0.530	0.855	1.313	1.546	1.701	1.817	1.909	2.048	2.154	2.237	2.368	2.467	2.548	2.673	2.763
29	0.256	0.530	0.854	1.311	1.544	1.699	1.814	1.906	2.045	2.150	2.233	2.364	2.462	2.543	2.667	2.756
30	0.256	0.530	0.854	1.310	1.543	1.697	1.812	1.904	2.042	2.147	2.230	2.360	2.457	2.537	2.661	2.750
40	0.255	0.529	0.851	1.303	1.532	1.684	1.796	1.886	2.021	2.123	2.203	2.329	2.423	2.501	2.619	2.704
50	0.255	0.528	0.849	1.299	1.526	1.676	1.787	1.875	2.009	2.109	2.188	2.311	2.403	2.479	2.594	2.678
75	0.254	0.527	0.846	1.293	1.517	1.665	1.775	1.861	1.992	2.090	2.167	2.287	2.377	2.450	2.562	2.643
100	0.254	0.526	0.845	1.290	1.513	1.660	1.769	1.855	1.984	2.081	2.157	2.276	2.364	2.436	2.547	2.626
∞	0.253	0.524	0.842	1.282	1.501	1.645	1.751	1.834	1.960	2.054	2.127	2.241	2.326	2.395	2.501	2.576