

ICT의 가치를 이끄는 사람들!!  
ICT 기술사의 사람들!!

123회

# 정보관리기술사 기출풀이 4교시

## 국가기술자격 기술사 시험문제

정보처리기술사 제 123 회

제 4 교시

분야	정보처리	종목	정보관리기술사	수험 번호		성명	
----	------	----	---------	-------	--	----	--

\* 다음 문제 중 4 문제를 선택하여 설명하시오. (각 25 점)

1. 최근 Netflix 등 CP(Contents Providers)들의 망관리 비용을 요구하는 법 제화 움직임이 일자 그들의 불만이 표시되고 있다. 4 차 산업혁명시대의 망중립 성 이슈에 대하여 설명하시오.
  - 가. 쟁점 사항 및 정책 이슈
  - 나. 시사점 및 대응 방안
2. A 발주기관은 B 수행업체에게 IT 아웃소싱업무를 위탁하였다. 다음에 대하여 설명하시오.
  - 가. 발주가능한 아웃소싱의 유형
  - 나. SLA(Service Level Agreement)와 OLA(Operational Level Agreements)
  - 다. Penalty/Incentive 관리체계, SIP(Service Improvement Plan), Annual Reset
3. 정보윤리와 관련하여 다음에 대하여 설명하시오
  - 가. 정보윤리 정의 및 원칙
  - 나. 컴퓨터(사이버) 범죄의 정의와 특징, 종류와 그 대응책
  - 다. 컴퓨터(사이버) 중독의 정의, 종류와 그 문제점
4. 자연어 처리와 관련하여 다음에 대하여 설명하시오.
  - 가. NER(Named Entity Recognition)과 Word Embedding
  - 나. BERT 와 GPT-3 의 비교

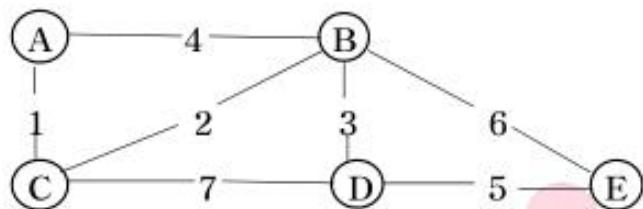
5. 정량적 위험분석의 정의와 필요성을 설명하고, 아래의 4 가지 위험분석 기법에 대하여 예시를 들어 설명하시오.

- 의사결정나무 분석(Decision Tree Analysis)
- 금전적 기대값 분석(Expected Monetary Value Analysis)
- 몬테카를로 시뮬레이션(Monte Carlo Simulation)
- 민감도 분석(Sensitivity Analysis)

6. 다음에 대하여 설명하시오.

가. 최단경로 알고리즘의 유형 4 가지

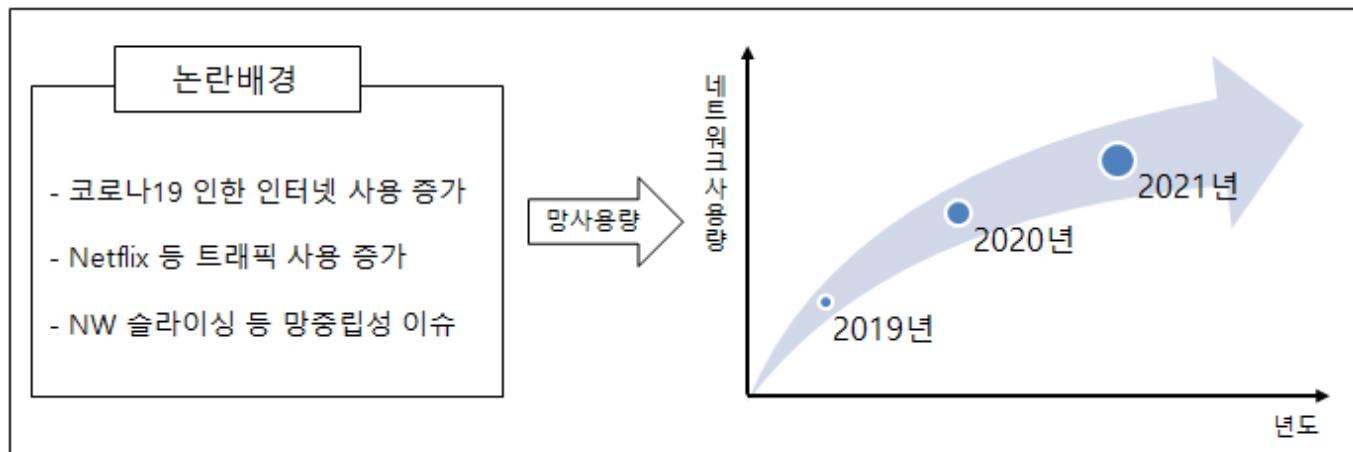
나. 다음 그래프에서 4 가지 알고리즘 계산방법



- 문제 1. 최근 Netflix 등 CP(Contents Providers)들의 망관리 비용을 요구하는 법제화 움직임이 일자 그들의 불만이 표시되고 있다. 4차 산업혁명시대의 망중립성 이슈에 대하여 설명하시오.
- 가. 쟁점 사항 및 정책 이슈  
나. 시사점 및 대응 방안

출 제 영 역	디지털서비스	난 이 도	★★★★☆
출 제 배 경	<ul style="list-style-type: none"> <li>- 코로나 19 이후 네트워크 사용량 증가로 SKBB와 넷플릭스간 법적 분쟁</li> <li>- 망 중립성 및 인터넷 트래픽 관리에 관한 가이드라인 개정(안)을 마련</li> </ul>		
출 제 빙 도	출제		
참 고 자 료	<ul style="list-style-type: none"> <li>- 망 중립성 논란과 향후 전망 (KDB 미래전략연구소)</li> <li>- 망중립성 규제에 관한 최근 논의 동향(ITFIND)</li> <li>- 망 중립성 및 인터넷 트래픽 관리에 관한 가이드라인 (과기부)</li> </ul>		
Key word	OTT, 코드커팅, 제로레이팅, 기울어진 운동장, 넷플릭스법, NW 슬라이싱, 5G, 캐시서버, 관리형 서비스		
풀 이	김준삼(118 회 정보관리기술사)		
감 수	공수재(111 회 컴퓨터시스템응용기술사)		

### 1. 망중립성 논란 배경



- 개념) 망 중립성은 통신사가 서비스 유형 또는 제공자 등에 따라 트래픽을 차별하는 것을 금지하는 원칙
- 코로나 19로 인해 집에 있는 시간이 늘어나면서 많은 양의 인터넷 트래픽이 발생함.
- 이로인해, 국내 ISP인 SK 브로드밴드는 이용자 증가로 인해 해외망 증설 등 유지 비용이 증가했다는 것을 이유로 넷플릭스 측에 망 이용료를 요구하여, SKBB와 넷플릭스간 법적 분쟁 및 망중립성 이슈 쟁점화

## 2. 쟁점 사항 및 정책 이슈

### 가. 망중립성 쟁점 사항

구 분	쟁점사항	설 명
CP 측면 (넷플릭스 등)	망중립성 유지	- CP는 망 접속료 증가, 콘텐츠 시장 경쟁 저해, ISP의 영향력 비대화를 우려하여 망 중립성 완화에 반대
	망 이용료 이중과금	- 넷플릭스 등 자신들의 서버가 속한 해외 ISP에 이미 정당한 사용료를 지불하고 있으므로, 국내 ISP에 망 사용료를 지불하는 것은 이중과중에 해당
ISP 측면 (SKBB 등)	망중립성 완화	- 5G 효율적 활용을 위하여 망 중립성 완화를 주장
	Mission Critical 서비스 차별 요구	- 자율주행차, 원격의료 등 민감 서비스에 대한 5G 속도 차별 허용 요구
	인프라 비용 분담	- 국내 ISP인 SK 브로드밴드는 이용자 증가로 인해 해외 망 증설을 네 차례나 시행하여 인프라 비용 분담 요구

- 국내 이동 3사 전체 트래픽의 32.8%가 구글, 넷플릭스가 차지하지만 망 사용료를 부담하지 않아 형평성 논란으로, 매년 급증하는 데이터 트래픽에 맞춰 통신망에 투자하려면 인터넷 업체들이 고통을 분담해야 한다고 주장

### 나. 망중립성 정책 이슈

구 분	정책이슈	설 명
인터넷 개념 재정의	규제 대상 검토	- 다양한 네트워크 품질을 요구하는 지능정보형 서비스의 출현으로 망중립성 규제 중심이 되는 인터넷 서비스 개념에 대한 검토 필요
	서비스의 확장	- 5G 확대로 인한 네트워크 슬라이싱 서비스의 효율적인 활용을 위한 서비스 확장 필요
	지능정보 서비스	- 자율주행자동차, 헬스케어 등 지능정보서비스 활용을 위한 검토 필요
관리형 서비스 검토	최선형 인터넷	- 다양한 관리형 서비스의 출현 및 경쟁관계 형성으로 개념의 정교화와 최선형(Best Effort) 망 보호이슈 발생할 것으로 예상
	망간 영향도 이슈	- 프리미엄 인터넷의 등장으로 망간 영향, 차별성 등 예상

- 넷플릭스법 제정으로 ISP는 글로벌 CP에게 망 이용료를 부과할 법적 근거 확보하였으나, 현실적으로 실효성이 낮을 것으로 예상

### 3. 시사점 및 대응방안

#### 가. 망중립성 시사점

구 분	기술요소	설 명
서비스 측면	망중립성 원화 전망	- 네트워크 슬라이싱 기술 도입 및 미국의 망중립성 원칙 폐지, 관리형 서비스에 대한 예외 인정 등 망중립성 원화 기조
	신규 사업모델 등장	- 생태계 내 제휴를 통해 다양한 사업모델 등장할 것으로 예상
	추가 요금 인상 및 고 퀄리티 서비스 제공	- 다양한 요금제로, 차별화된 서비스 및 다양한 컨텐츠로 고객 만족
비즈니스 측면	글로벌 CP 기업 매출 증대 및 영향력 확대	- 지속적으로 증가하는 OTT 요구사항으로 인해 글로벌 CP 기업 매출 증대 및 영향력 확대
	네트워크 사용량 지속 증가	- 언제 어디서나 사용가능한 인터넷 및 OTT 와 생활 패러다임이 가정으로 변하여 네트워크 사용량 지속 증가 예상

- ISP는 넷플릭스를 향해 "공짜 무임승차를 더 이상 방치할 없다"며 망 이용 대가를 요구하고 있고, 넷플릭스는 이 요구가 망중립성 원칙에 위배된다며 한국 법원에 소송까지 제기하는 등 첨예한 대립을 이어가고 있어 대응방안 마련 필요

#### 나. 망중립성 이슈 대응방안

구 분	대응방안	설 명
기술적 측면	캐시 서버 적용	- 캐시 서버 적용시 트래픽이 이동하는 물리적 거리가 줄어들어 ISP는 사용자에게 더 빠른 인터넷 서비스를 제공
	네트워크 슬라이싱 적용	- 하나의 물리적 네트워크를 복수의 상호 독립된 논리적 네트워크로 분할하여, 서로 다른 접속서비스를 요구하는 서비스들에게 차등화된 접속서비스를 제공하는 기술
관리적 측면	관리형 서비스	- 일반적으로 통용되는 최선형 인터넷의 제공 방식과 다른 트래픽 관리기술 등을 통해 전송 대역폭 등 트래픽 전송 품질을 보장하는 서비스
	제로레이팅	- 이용자가 특정 콘텐츠(앱)을 이용할 때 발생하는 데이터 요금을 부과하지 않은 서비스
	불공정 행위 요소 제재	- 시장의 지속적인 모니터링으로 사업자 간 불공정 행위 요소를 감시하고 제재하는 것이 필요

- 과기부는 2020년 12월 5G 등 네트워크 기술발전에 대응하기 위해 각계 의견을 수렴하여 망 중립성 및 인터넷 트래픽 관리에 관한 가이드라인 개정(안)을 마련함

#### 4. 망 중립성 및 인터넷 트래픽 관리에 관한 가이드라인

개정내용	설명
망중립 예외 조건 명확화	<ul style="list-style-type: none"> <li>- 망중립 예외서비스 제공요건을 명확히 위해, 해외 같이 특수 서비스 개념 도입</li> <li>- 특수서비스 개념 도입으로 시장의 불확실성 해소 기대</li> </ul>
특수서비스 제공조건 구체화	<ul style="list-style-type: none"> <li>- 특수서비스가 제공될 경우에도 일반 이용자가 이용하는 인터넷 품질은 적정수준으로 유지하는 등 남용 가능성을 차단</li> <li>- 통신사업자가 서비스 품질 적정수준 유지, 지속적 고도화, 특수서비스를 망중립성 원칙 회피 목적으로 제공하는 것을 금지</li> </ul>
투명성 강화	<ul style="list-style-type: none"> <li>- 통신사업자와 콘텐츠사업자 등 이용자 간 정보비대칭성 완화를 위해 투명성 강화</li> <li>- 통신사의 정보공개대상을 확대하고, 정부가 인터넷 접속서비스 품질 등을 점검, 관련 자료제출을 통신사에 요청할 수 있도록 근거 마련</li> </ul>
<ul style="list-style-type: none"> <li>- 망중립성 시장상황 등을 지속적으로 모니터링하여 인터넷 생태계의 건전하고 지속 가능한 발전을 도모하고 사업자 간 공정경쟁여건과 이용자의 권리를 보장하기 위한 정책적 노력 필요함</li> </ul>	

"끝"



#### 기출풀이 의견

1. 망중립성은 다수 기출되어 OTT, 코드커팅, 제로레이팅 등 많은 키워드 작성 중요
2. 기존의 망중립성은 찬성, 반대 입장이나, 이번에는 CP사와의 망관리에 관한 이슈이므로 포커스를 넷플릭스와 ISP 간의 이슈에 대해서 작성 필요

2. A 발주기관은 B 수행업체에게 IT 아웃소싱업무를 위탁하였다. 다음에 대하여 설명하시오.

가. 발주가능한 아웃소싱의 유형

나. SLA(Service Level Agreement)와 OLA(Operational Level Agreement)

다. Penalty/Incentive 관리체계, SIP(Service Improvement Plan), Annual

**Reset**

출 제 영 역	소프트웨어공학	난 이 도	★★★★☆
출 제 배 경	2021년 공공 IT 사업 규모 약 6조, 11.8% 증가하였고, SW에 신기술(클라우드, AI, 빅데이터, 블록체인 등) 적용 안정화 필요		
출 제 빈 도	출제		
참 고 자 료	<ul style="list-style-type: none"> <li>- 서비스수준협약(SLA) 기술동향 (전자통신동향분석)</li> <li>- 공공기관에서의 서비스 수준 협약(SLA) (Korea Science)</li> <li>- 서비스 수준 관리(SLM) 문서 정보 기술(정보 기업)</li> </ul>		
Key word	일괄적 아웃소싱, 선택적 아웃소싱, 서비스 가동률(%), 동일장애 발생률(%), 장애 및 오류건수, SR 적기 처리율(%), 네트워크 가동률(%), 네트워크 장애건수		
풀 이	김준삼(118회 정보관리기술사)		
감 수	공수재(111회 컴퓨터시스템응용기술사)		

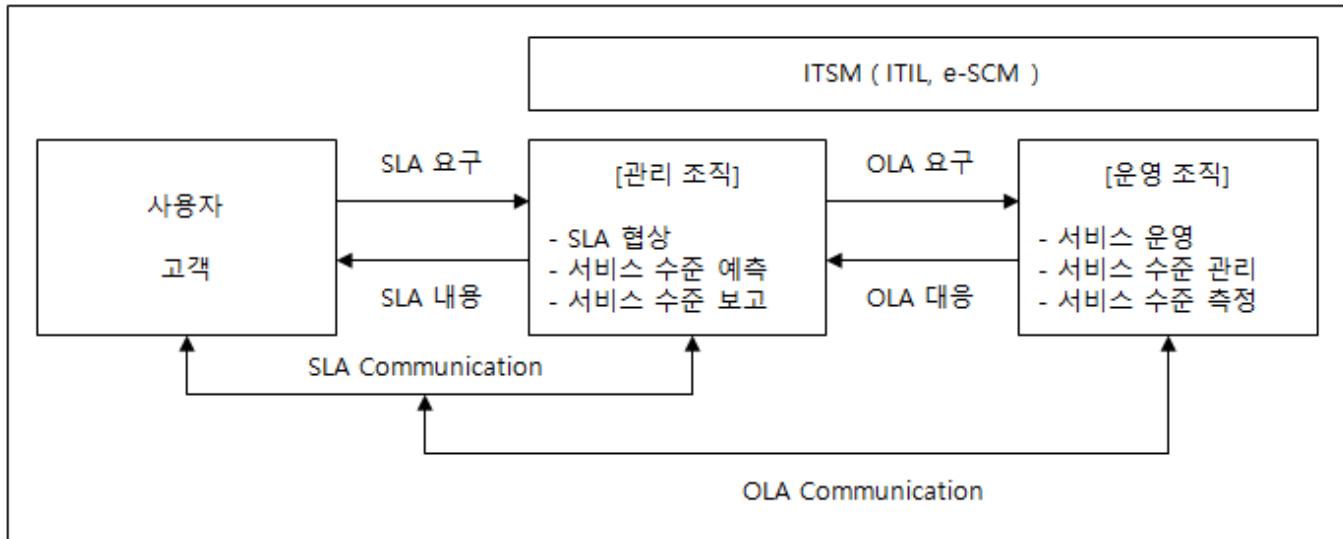
### 1. 외부의 전문기업에 위탁 처리, 아웃소싱의 유형

구 분	일괄적 아웃소싱	선택적 아웃소싱
개념	<ul style="list-style-type: none"> <li>- 업무 또는 기능에 대하여 일괄적으로 하나의 업체에게 위탁하는 형태의 아웃소싱</li> </ul>	<ul style="list-style-type: none"> <li>- 업무를 전문성이 있는 업체들에게 분할하는 아웃소싱</li> </ul>
장점	<ul style="list-style-type: none"> <li>- 하나의 업체로의 책임소재의 명확성</li> <li>- 의사소통 창구 일원화</li> </ul>	<ul style="list-style-type: none"> <li>- 영역별 전문성 확보로 품질이 좋은 서비스 개발</li> </ul>
단점	<ul style="list-style-type: none"> <li>- 서비스 제공 업체에 기술적 종속 우려</li> </ul>	<ul style="list-style-type: none"> <li>- 하지만 반대로 밴더들이 많기 때문에 책임소재의 불명확성</li> <li>- 시스템의 일관성 및 통합성 유지 어려움</li> </ul>

- 개념 : 조직의 전략적 목표를 달성하기 위하여 자산의 이전을 포함한 정보시스템 요소의 일부 또는 전부를 외부전문업체에 위탁하여 운영하게 하는 장·단기 계약

## 2. SLA 와 OLA 설명

### 가. SLA 와 OLA 관계



- SLA 와 OLA 는 중점사항은 다르나, 모두 프로젝트를 성공적으로 완료하려고 시도하기 때문에 궁극적으로 둘의 목표는 동일함.

### 나. SLA 와 OLA 설명

구 분	SLA	OLA
개념	- IT 서비스를 제공하는 업체와 사용하는 업체간의 서비스 계약서	- 내부 조직 간 의사소통 및 관리를 효율적으로 하기 위해 서비스 공급자 내부 부서간에 교환하는 합의서
중점사항	- 계약의 서비스 부분에 중점	- 유지 관리 및 기타 서비스에 관한 계약
계약대상	- 서비스 제공 업체-최종 사용자 계약	- 내부 계약

- 기업은 SLA 또는 OLA 를 준비하기 전에 충분한 시간을 보내고 모든 관련 요소를 고려함으로써, 예상 된 결과가 달성되지 않아 Penalty 와 같은 불이익 대응

## 3. Penalty/Incentive 관리체계, SIP, Annual Reset

### 가. Penalty/Incentive 관리체계 설명

구 분	Penalty	Incentive
개념	- 서비스 최소 수준을 달성하지 못했을 경우 서비스 제공자가 사용자에게 지급하는 금액 또는 서비스 개선활동	- 서비스 목표 수준을 초과 달성한 경우에 사용자가 서비스 제공자에게 제공하는 보상
조건	- 인력 교체	- 회식비

- 실제 프로젝트에서 SLA 지표 달성을 기본이므로, Incentive 는 거의 없고 Penalty 만 존재함.

## 나. SIP, Annual Reset 설명

구 분	SIP (Service Improvement Plan)	Annual Reset
개념	- 주기별로 측정된 Service 결과에 대한 개선 사항들을 작성한 계획	- IT 환경 변화를 고려하여, 발주자와 계약당사자간 매년 서비스 목표 수준 재설정
특징	- 서비스 검토회의의 결과 조치 및 개선이 필요한 경우 SIP를 통해 사용자 교육, 시스템 테스트, 문서화 등을 포함하며 원활한 서비스 개선활동을 할 수 있도록 별도의 예산을 편성	- 목표 수준은 서로 동의하는 객관적이고, 합리적인 근거에 의해 협의 - 단, 서로 협의하에 기준 서비스 목표 수준 지속 가능함.
<ul style="list-style-type: none"> <li>- 지나치게 계약 기간을 길 경우 부정적으로 보는 경향이 있어, IT 환경의 변화를 고려하여, 매년 1회 계약을 갱신함.</li> </ul>		

## 4. SLA 평가 지표와 측정 방법

구 분	평가지표	측정방법
하드웨어	서비스 가동률(%)	<ul style="list-style-type: none"> <li>- 서비스 제공자가 서비스 시간 동안에 제공하는 가용성 목표치</li> <li>- 서비스 가동률(%) = <math>(1 - \text{장애시간}/\text{서비스시간}) * 100</math></li> </ul>
	동일장애 발생률(%)	<ul style="list-style-type: none"> <li>- 이미 발생했던 장애와 동일한 장애가 재발한 비율</li> <li>- 동일장애발생률(%) = <math>(\text{동일 장애 발생건수} / \text{총 장애 발생건수}) * 100</math></li> </ul>
소프트웨어	장애 및 오류건수	<ul style="list-style-type: none"> <li>- 장애 및 오류 10 건 이하</li> </ul>
	SR 적기 처리율(%)	<ul style="list-style-type: none"> <li>- 요청한 완료일 이내에 서비스를 제공해준 비율</li> <li>- SR 적기 처리율(%) = <math>(\text{완료 예정일자 이내에 완료한 서비스 요청건수} / \text{측정기간에 완료 예정인 서비스 요청건수}) * 100</math></li> </ul>
네트워크	네트워크 가동률(%)	<ul style="list-style-type: none"> <li>- <math>(1 - \text{장애시간}/\text{네트워크 가동시간}) * 100</math></li> </ul>
	네트워크 장애건수	<ul style="list-style-type: none"> <li>- 장애 건수 3 건 이하</li> </ul>

- 기존 SLA 통해 상호간 의사소통의 오해를 없애고, 공급업체에 의해 제공되는 서비스 품질의 객관적 평가 가능
- 또한 시스템 자원 및 인원에 대한 적절한 통제 및 평가 가능함

"끝"

## 기출풀이 의견

1. 대부분 SLA 문제는 클라우드 SLA 관련되었으나, 이번 문제는 프로젝트 관련 SLA 문제임
2. 물어보는게 많아 작성하기 힘든 문제였으나, 반드시 정확한 개념은 작성되어야 고득점 가능함.

3. 정보윤리와 관련하여 다음에 대하여 설명하시오.

문 제

가. 정보윤리 정의 및 원칙

나. 컴퓨터(사이버) 범죄의 정의와 특징, 종류와 그 대응책

다. 컴퓨터(사이버) 중독의 정의, 종류와 그 문제점

출 제 영 역	디지털서비스	난 이 도	★★★★☆
출 제 배 경	인공지능의 발달로 사이버 범죄 고도화 및 지능화되고, 개인화된 환경 및 언제 어디서든 컴퓨터 활용 증가에 따른 중독 증가 현상 발생		
출 제 빈 도	미출제		
참 고 자 료	<ul style="list-style-type: none"> <li>- 사이버 정보 윤리 (충청북도교육청)</li> <li>- 사이버범죄예방을 위한 법제도적 해결방안 (사이버범죄 특별법)</li> <li>- 사이버 1388 청소년상담센터</li> <li>- 인터넷 중독, 진단과 치료 (한림대학교)</li> </ul>		
Key word	존중, 책임, 정의, 해약금지, 비대면성, 익명성, 시간/공간의 무제한성, 경찰청 사이버캅 앱, 셋다운제, PC 제한 프로그램		
풀 이	김준삼(118 회 정보관리기술사)		
감 수	공수재(111 회 컴퓨터시스템응용기술사)		

## 1. 정보윤리 정의 및 원칙

구 분	원 칙	설 명
정의	- 정보 사회의 구성원으로 옳고 그름, 좋음과 나쁨, 윤리적인 것과 비윤리적인 것을 올바르게 판단하고 행동하는 데 필요한 규범적 기준 체계	
원 칙	존중(respect)	- 자신에 대한 존중은 자신의 생명과 몸을 본래적 가치를 지닌 것으로 대우할 것을 요구하는 것이며, 타인에 대한 존중은 타인의 지적 재산권, 프라이버시, 다양성을 인정하고 존중하는 것을 의미
	책임(responsibility)	- 사이버공간에서 정보 이용자 및 정보 제공자는 자신의 행동이 어떤 결과를 가져오게 될 것인지에 대해 미리 심사숙고해야 할 필요
	정의(justice)	- 사이버 공간에서 각자는 자신이 제공하는 정보의 진실성, 비편향성, 완전성, 공정한 표현을 추구해야 하며 타인의 기본적 자유와 권리를 침해하지 않아야 함
	해약금지(non-maleficence)	- 사이버 성폭력, 크래킹, 바이러스 유포 등과 같은 행위들은 타인에게 명백하게 해로움을 주는 것이므로 마땅히 지양해야 할 행동(남에게 피해를 주면 안됨)

- IT 강국으로 핸드폰 뿐만 아니라 빠른 네트워크 사용 및 언제 어디서나 인터넷을 사용 가능
- 이로인해 많은 사람들이 무문별하게 인터넷을 사용하거나 이를 범죄에 이용하는 경우가 많이 발생하는 사례가 증가됨

## 2. 컴퓨터(사이버) 범죄의 정의와 특징, 종류와 그 대응책

### 가. 컴퓨터(사이버) 범죄의 정의와 특징

구 분	특 징	설 명
정의	- 사이버공간에서 행하여지는 모든 범죄적 현상, 즉 사이버 공간을 범행의 수단, 대상 혹은 무대로 삼는 범죄현상	
특징	비대면과 익명성	- 신분노출의 위험성이 낮아 대담한 범죄를 저지르고 가해자를 찾기도 어려움
	전문성과 기술성	- 프로그램 조작 등 컴퓨터 관련 전문적인 지식과 기술이 요구됨
	시간·공간의 무제한성	- 시간과 공간의 제약이 없어 범죄 행위 또한 언제 어디서나 발생 가능

- 사이버 범죄는 인터넷발달로 피해자의 연령대가 청소년에서부터 장년층까지 다양하게 분포되어 있으며, 그 범죄유형도 로맨스 스캠, 몸캠파싱, 디지털 성범죄, 피싱 사기, 스미싱, 이메일 무역 사기, 사이버 명예훼손, 사이버 모욕죄 등 여러 가지 유형 존재함

### 나. 컴퓨터(사이버) 범죄의 종류와 대응책

구분	범죄의 종류와 대응책	범죄의 종류와 대응책 설명
사이버 범죄 종류	테러형 범죄	- 해킹, DoS, 랜섬웨어, 악성프로그램, 바이러스 제작 및 유포, 메일폭탄, 피싱, 파밍, 스미싱
	일반형 범죄	- 사이버 사기, 불법복제, 불법/유해사이트, 사이버스토킹, 모욕죄, 성폭력, 개인정보침해, 명예훼손죄, 협박/공갈
사이버 범죄 대응책	관리적 대응	- OS 보안패치, 정보윤리·보안 교육, 정품 SW 사용, 비인가 사이트 접속 금지, 네티켓, 비밀번호 주기적 변경
	물리적 대응	- 보안 USB, 망분리, 중고물품 직거래,
	기술적 대응	- 백신, 전자서명, 방화벽, 스크러빙센터, 워터마킹, 웹사이트 인증, SIEM, IPS, IDS, EDR, SNI 필터링

- 컴퓨터 범죄의 대응으로 경찰청에서 개발한 어플 '경찰청 사이버캅'은 악성 발신자, 스팸, 보이스피싱 등 전화번호 확인, 사이버 범죄 예방에 대한 PUSH 기능 등 제공함.

## 3. 컴퓨터(사이버) 중독의 정의, 종류와 그 문제점

### 가. 컴퓨터(사이버) 중독의 정의와 종류

구 分	종 류	설 명
정의	지나치게 컴퓨터를 사용하여 일상생활에 심각한 사회적, 정신적, 육체적 및 금전적 지장을 받고 있는 상태	

종류	사이버 관계 중독(SNS)	- 인터넷 동호회, 미니홈피, 개인홈페이지 등을 만들어 과도한 시간을 사용함
	사이버 섹스 중독	- 가상공간에서 성적 대화를 나누거나 포르노 영상을 감상하는데 집착함
	게임 중독	- 다양한 종류의 게임을 오랜 시간 즐기는 것과 함께 지나친 아이템을 구입함

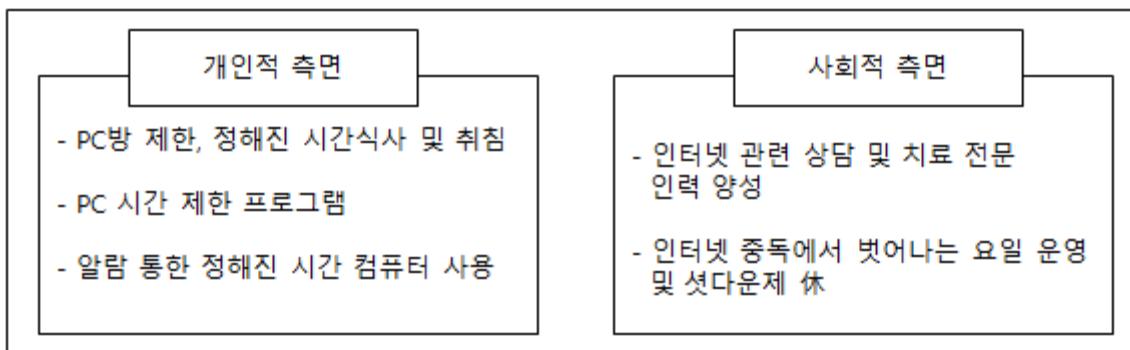
- 컴퓨터 중독은 만성 피로감, 음란물과 비방/욕설/언어적 폭력, 가상공간과 현실 사이의 구분이 모호해져 일탈행동을 야기함.

#### 나. 컴퓨터(사이버)중독의 문제점

구 분	문제점	설 명
심리적 측면	대인기피증	- 온라인의 높은 유대관계로 인해 현실의 대인관계가 줄어드는 문제
	주의력 결핍	- 주의력이 떨어져 산만하고 ADHD를 발생시키는 원인이 됨
	수면장애	- 불규칙하고 늦은 취침시간으로 인해 수면장애 발생
	폭력성	- 과잉행동 및 충동성에 따른 공격적이고 반항심 유발
사이버 범죄 측면	개인정보유출	- 사회공학 기법, 불법 계정을 통해 DB의 개인정보 불법 유출
	인터넷 사기	- 피싱, 파밍을 통해 금융 계좌정보, 계정 정보 탈취와 금품 갈취
	사이버 폭력	- 악성 댓글, 허위사실 유포 등으로 특정인 명예훼손 및 괴롭힘

- 컴퓨터 중독은 개인적 노력과 사회적인 노력이 모두 뒤따라야 하는데, 한 가지 알아야 할 것은 컴퓨터 중독은 예방이 우선되어야 하므로, 중독예방 교육 확대가 필요함.

#### 4. 컴퓨터(사이버)중독 대응방안



- 특히 일주일에 한번씩, 인터넷 중독에서 벗어나는 요일 및 셧다운제 운영을 통해, 가족과 함께 여가를 보낼 수 있는 여러 가지 활동 시도 필요함.

#### 기출풀이의견

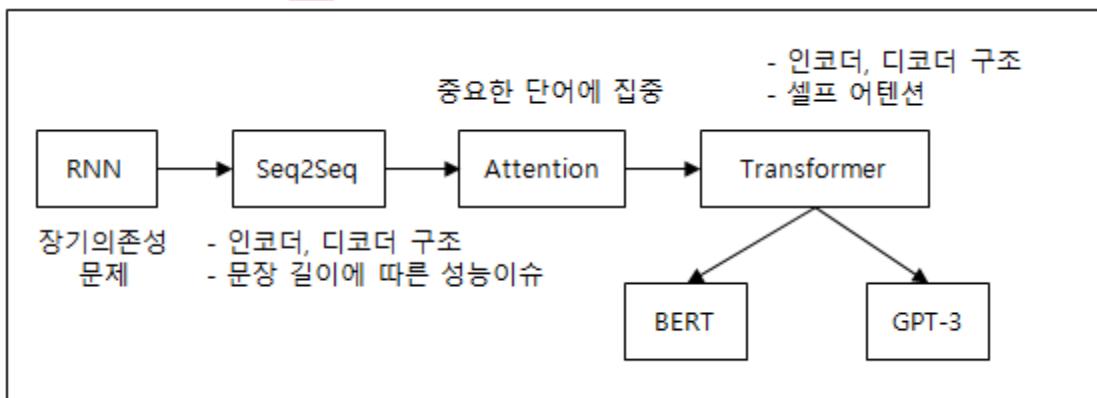
- 정보윤리 원칙 작성성이 어려웠으나, 다른 문제들이 어려워 대부분 작성했을 것이라 예상
- 질문이 많아 글씨가 많기 때문에, 중요한 부분엔 반드시 밑줄을 통해 강조하여 가독성 확보하여야 함.

#### 4. 자연어처리와 관련하여 다음에 대하여 설명하시오.

- 문제  
가. NER(Named Entity Recognition)과 Word Embedding  
나. BERT 와 GPT-3 의 비교

출제영역	디지털서비스	난이도	★★★★☆
출제배경	코로나 19 이후, 디지털 사회로의 패러다임 전환되고, 언어모델중 우수한 성능을 보이는 모델 등장		
출제빈도	미출제		
참고자료	<ul style="list-style-type: none"> <li>- BERT 설명 발표 자료 (Banana Media Lab)</li> <li>- 딥러닝 자연어처리 - RNN에서 BERT 까지(SlideShare)</li> <li>- GPT3 - Language Models are Few(Shot Learners)</li> </ul>		
Keyword	Fine tuning, few shot learning, Transformer, Attention, GloVe, FastText, word2vec, CBOW, SKIPGRAM		
풀이	김준삼(118회 정보관리기술사)		
감수	공수재(111회 컴퓨터시스템응용기술사)		

#### 1. 자연어 처리 개요



- 자연어 처리는 컴퓨터가 인간의 언어를 이해하고 해석하며 조작하도록 돋는 인공지능의 한 분야
- 최초 RNN 기반의 Seq2Seq 언어모델이 많이 사용 되었으나, 문장이 길어질 경우 장기의존성 문제가 발생
- Attention 매커니즘 통해 최신의 언어모델인 Transformer, BERT, GPT-3 등 등장함.

#### 2. NER(Named Entity Recognition)과 Word Embedding

##### 가. NER(Named Entity Recognition) 설명

구분	설명
개념	- 미리 정의해 둔 사람, 회사, 장소, 시간, 단위 등에 해당하는 단어(개체명)를 문서에서 인식하여 추출 분류하는 기법
개체명 종류	- 인명(person), 지명(location), 기관명(organization), 시간(time) 등으로 분류

활용	- 정보 추출, 자연어 처리, 정보 검색, 챗봇 등
사례	<p>[Input] 헤이카 x 오, 배달의 x 족에서 교 x 치킨 주문해줘~</p> <p>[NER 분류]</p> <ul style="list-style-type: none"> <li>- 헤이카 x 오 : 본인의 개체명</li> <li>- 배달의 x 족 : 앱명</li> <li>- 교 x 치킨 : 브랜드명</li> </ul> <p>→ 이 외 가격, 메뉴, 주문시간 등 다양한 개체명이 추가</p>

- NER 즉 개체명 인식을 하기 위해, 컴퓨터가 이해할 수 있도록 Word Embedding 이 기반이 되어야 함.

#### 나. Word Embedding 설명

구 분	종류	설 명
정의	컴퓨터가 이해할 수 있도록 텍스트를 벡터(수치화)화 하는 기법	
개념도	<p>word</p> <p>vector</p>	
유형	<p><b>Word2Vec</b></p> <ul style="list-style-type: none"> <li>- 구글에서 만든 추론 기반의 단어를 벡터화하는 알고리즘</li> <li>- 종류 : CBOW, SKIP-GRAM</li> </ul> <p><b>GloVe</b></p> <ul style="list-style-type: none"> <li>- stanford에서 개발한 워드임베딩 알고리즘</li> <li>- 기존의 카운트 기반의 LSA(Latent Semantic Analysis)와 예측 기반의 Word2Vec 의 단점을 지적하며 이를 보완한다는 목적으로 등장.</li> </ul> <p><b>FastText</b></p> <ul style="list-style-type: none"> <li>- 페이스북에서 개발한 워드임베딩 알고리즘</li> <li>- 원래 단어를 부분단어(subword)의 벡터들로 표현한다는 점을 제외하고는 Word2Vec과 거의 유사</li> </ul>	

- 텍스트를 수치화 하는 기법중 One Hot Encoding 은 나름대로 좋은 성능을 내지만 단어의 의미 또는 개념 차이를 벡터안에 담지 못한다는 큰 단점이 존재함.

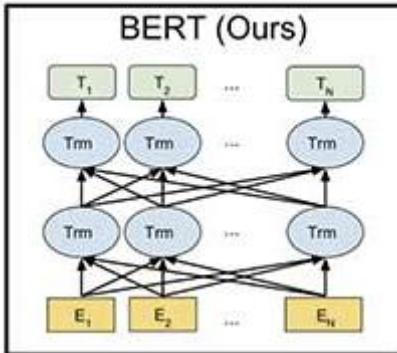
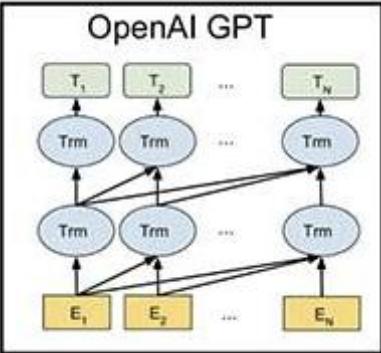
### 3. BERT 와 GPT-3의 비교

#### 가. BERT 와 GPT-3의 개념 비교

구 분	BERT	GPT-3
개념	트랜스포머의 인코더 구조로 구성되어 file turning 된 양방향 자연어 처리 모델	트랜스포머의 디코더 구조로 구성되어 few shot Learning 된 순방향 자연어 처리 모델

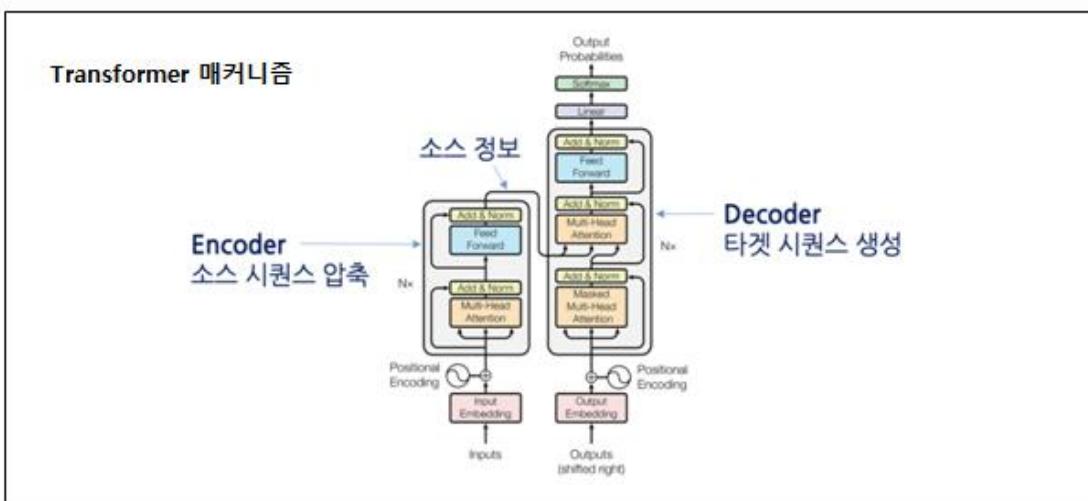
- 기존 BERT 는 문장 중간에 빈칸을 만들고 해당 빈칸에 어떤 단어가 적절할지 맞추는 모델이고, GPT-3 는 이전 단어들이 주어졌을 때 다음 단어가 무엇인지 맞추는 언어 모델
- 즉, GPT 는 문장 생성에, BERT 는 문장의 의미를 추출하는 데 강점을 지님

## 나. BERT 와 GPT-3 의 상세 비교

구 분	BERT	GPT-3
개념도		
개발주체	Google	OpenAI
모델구조	트랜스포머의 Encoder 만 사용	트랜스포머의 Decoder 만 사용
주요특징	fine tuning	few shot Lerning
핵심기능	문장의 의미 추출	문장 생성
사례	어제 카페 갔었어 <input type="text"/> 사람 많더라	어제 카페 갔었어 거기 사람 많더라

- BERT 와 GPT-3 가 성능을 가지고 경쟁하는 것처럼 보이지만 실제 Transformer 로 부터 Encoder 및 Decoder 부분만 사용하는 언어모델 임.

## 4. BERT 와 GPT-3 의 태생, Transformer 설명



- 2017년 구글이 발표하여 attention 매커니즘을 사용하여 인코더, 디코더로 구현된 언어처리 모델로 최근 딥러닝 언어모델 Transformer에서 파생됨.

## 기출풀이의견

- 최근에 발표한 인공지능 언어모델에 대한 문제로 작성하기 쉽지 않을 것으로 예상됨
- BERT, GPT-3는 자연어처리 쪽에서 매우 우수한 성능을 보이는 기술이므로 추가 학습 필요함.

5. 정량적 위험분석의 정의와 필요성을 설명하고, 아래의 4 가지 위험분석 기법에 대하여 예시를 들어 설명하시오.

- 문제
- 의사결정나무 분석(Decision Tree Analysis)
  - 금전적 기대값 분석(Expected Monetary Value Analysis)
  - 몬테카를로 시뮬레이션(Monte Carlo Simulation)
  - 민감도 분석(Sensitivity Analysis)

출제영역	디지털서비스	난이도	★★★★☆
출제배경	117회 정보관리기술사 문제와 동일한 문제로, 프로젝트에서의 위험관리의 중요성 확보 필요		
출제빈도	출제		
참고자료	<ul style="list-style-type: none"> <li>- 위험관리, 어떻게 할 것인가 - LG 경제연구원</li> <li>- 프로젝트 위험관리 (ePMFORUM)</li> <li>- 프로젝트 위험 관리 도구 및 기법 (PMO)</li> </ul>		
Keyword	효용이론, EMV 계산, EMV = 발생확률(Probability) X 가치(value), EMV > 0 : 기회, EMV < 0 : 위험, 확률분포, 난수, 토네이도		
풀이	김준삼(118회 정보관리기술사)		
감수	공수재(111회 컴퓨터시스템응용기술사)		

### 1. 정량적 위험분석의 정의와 필요성

구분	필요성	설명
정의	- 식별된 위험의 발생가능성 및 영향력을 수치화하는 분석 기법	
필요성	비즈니스 빠른 변화 대응	- 소프트웨어 수익 모델의 변화, 플랫폼 비즈니스 활성화, IT 인프라 시장의 변화
	불확실성 증가	- 요구사항 변화, 요구사항 불확실성으로 인한 위험 대응 필요
	복잡성 증가	- IT에 대한 의존도 증가와 SW의 급변하는 기술, AI, 블록체인, 빅데이터 등에 대한 기술 습득 어려움
<ul style="list-style-type: none"> <li>- 급변하는 비즈니스 환경에서 프로젝트 완수에 부정적 요소의 식별, 통제, 대응을 최소화하기 위해 정량적 위험분석 필요함.</li> <li>- 이로인해 비즈니스 연속성 및 위험에 대한 사전예측, Issue 대응으로 비용 최소화 가능</li> </ul>		

## 2. 의사결정나무 분석과 금전적 기대값 분석 설명

### 가. 의사결정나무 분석 설명

구 분	이론	설 명
정의	- 의사 결정에 대한 기대값(EMV)을 계산하여, 이를 의사결정에 반영하는 위험관리 기법	
특징	효용이론	- 위험에 대한 선호도에 따라 의사결정 결과가 달라질수 있다는 이론
예시	<p style="text-align: right;">EMV (기대값)</p>	
결과	<ul style="list-style-type: none"> <li>- EMV 값이 신축 일경우 80 억이므로, "신축" 선택</li> <li>- 단, 효용이론에 근거하여 리스크를 최소화하기 위해 "개선" 선택도 가능함.</li> </ul>	
	<ul style="list-style-type: none"> <li>- 의사결정나무 분석에서 금전적 기대값 분석(EMV)를 통해 의사결정 활용</li> </ul>	

### 나. 금전적 기대값 분석 설명

구 분	설 명
정의	- 불확실성을 고려하여 사건의 가치와 발생 확률을 곱해 미래의 기대값을 계산하는 기법
분석방법	<ul style="list-style-type: none"> <li>- <math>EMV = \text{발생확률}(\text{Probability}) \times \text{가치}(\text{value})</math></li> <li>- <math>EMV &gt; 0</math> : 기회, <math>EMV &lt; 0</math> : 위험</li> </ul>
예시	<p>예) 복권 1장에 1,000 원, 당첨금액 10,000 원, 당첨확률 7% 기대화폐가치(EVM)는 얼마인가?</p> $\begin{aligned} EMV &= \text{특정 Case 기대값} + \text{반대 Case 기대값} - \text{기회비용} \\ &= (10,000 * 0.07 - 1,000) + (0 * 0.93 - 1,000) \\ &= -300 \end{aligned}$
결과	<ul style="list-style-type: none"> <li>- 결과값이 음수(-300) 이므로 복권 1장을 사는 것은 당첨확률이 적음.</li> </ul>
	<ul style="list-style-type: none"> <li>- EMV가 양수 값으로 표현되면 기회, 음수값으로 표현되면 리스크도 판단하여 의사결정에 활용</li> </ul>

### 3. 몬테카를로 시뮬레이션과 민감도 분석 설명

#### 가. 몬테카를로 시뮬레이션 설명

구 분	설 명								
정의	- 확률분포를 가정하여 모델을 만들고 난수를 발생시켜 일정/원가측면의 준수확률을 계산하는 기법								
분석방법	- 난수 발생 ▶ 현재 분석하려는 발생 이벤트로의 변환 ▶ 한 번의 이벤트에 대한 결과 획득 ▶ 앞의 과정을 N 번 반복하여 확률 데이터 획득 ▶ 결과(기대치) 획득								
예시	<ul style="list-style-type: none"> <li>- A 모듈 설계시 최소 12~14 일 소요됨</li> <li>- 소요기간 12~14 일중 난수를 발생시켜 랜덤 선택</li> </ul> <div style="text-align: center;"> <table border="1" style="margin-bottom: 10px;"> <tr> <th>일정</th> <th>횟수</th> </tr> <tr> <td>12</td> <td>2</td> </tr> <tr> <td>13</td> <td>6</td> </tr> <tr> <td>14</td> <td>2</td> </tr> </table> </div>	일정	횟수	12	2	13	6	14	2
일정	횟수								
12	2								
13	6								
14	2								
결과	<ul style="list-style-type: none"> <li>- A 모듈 설계시 완료예정일을 13 일로 선택하여 안정적 프로젝트 진행</li> <li>- 어떤 상황이 정해져 있거나, 변수의 관계가 확실하여 예측치를 정확하게 찾을 수 있는 확정모형과는 달리, 일반적으로 많은 부분은 고정되어 있지않고, 상황이 급변하기 때문에 결과를 정확하게 예측하기 어려움.</li> <li>- 이러한 예측 불가능한 확률모형에서는 분석적 방법으로 해를 찾는 것이 불가능하기 때문에, 일련의 난수를 반복적으로 발생시켜 시뮬레이션을 통해 답을 찾을 수 있음</li> <li>- </li> </ul>								

#### 나. 민감도 분석 설명

구 분	설 명																		
정의	- 위험이 프로젝트에 미치는 영향을 분석하여 영향력이 큰 순서대로 배열하는 기법 (토네이도 기법)																		
예시	<table border="1" style="margin-top: 10px; width: 100%;"> <tr> <th>영향 요인</th> <th>영향력 (양수)</th> <th>영향력 (음수)</th> </tr> <tr> <td>소비자 구매력 변동</td> <td>900억</td> <td>-900억</td> </tr> <tr> <td>원자재가 변동</td> <td>800억</td> <td>-650억</td> </tr> <tr> <td>특허 사용료</td> <td>350억</td> <td>-350억</td> </tr> <tr> <td>환경 규제 대응</td> <td>120억</td> <td>-200억</td> </tr> <tr> <td>사업적 가치 총계</td> <td>900억, 800억, 350억, 120억</td> <td></td> </tr> </table>	영향 요인	영향력 (양수)	영향력 (음수)	소비자 구매력 변동	900억	-900억	원자재가 변동	800억	-650억	특허 사용료	350억	-350억	환경 규제 대응	120억	-200억	사업적 가치 총계	900억, 800억, 350억, 120억	
영향 요인	영향력 (양수)	영향력 (음수)																	
소비자 구매력 변동	900억	-900억																	
원자재가 변동	800억	-650억																	
특허 사용료	350억	-350억																	
환경 규제 대응	120억	-200억																	
사업적 가치 총계	900억, 800억, 350억, 120억																		

결과	<ul style="list-style-type: none"> <li>- 소비자 구매력 변동, 원자재가 변동, 특히 사용료 순으로 우선순위가 높으면 이에 따른 집중 관리 계획 필요함.</li> </ul>
	<ul style="list-style-type: none"> <li>- 민감도 분석은 영향력이 높은 것부터 위에서 아래로 표시하고, 이때 막대 그래프 크기가 클수록 영향력이 크므로 우선순위가 높아 우선적으로 관리해야 함.</li> </ul>

#### 4. 위험 대응 계획 수립

구 분	대응 계획	설 명
위험에 따른 전략	<b>회피 (Avoid)</b>	<ul style="list-style-type: none"> <li>- 변경, 제거, 목표자체 변경, 상황 자체에 처하지 않는 것</li> </ul>
	<b>전가 (Transfer)</b>	<ul style="list-style-type: none"> <li>- 리스크의 영향, 대응, 책임을 제 3 자에게 넘기는 것</li> <li>- 책임을 넘기는 것이지 불확실성은 사라지지 않는다.</li> </ul>
	<b>완화 (Mitigate)</b>	<ul style="list-style-type: none"> <li>- 리스크 발생의 확률 및 영향력을 허용 가능한 한계선까지 낮추는 것</li> </ul>
	<b>수용 (Accept)</b>	<ul style="list-style-type: none"> <li>- 대응 전략을 수립하기 어려운 경우 리스크를 안고 가는 것</li> <li>- 발생 시 확인</li> </ul>
기회에 따른 전략	<b>활용 (Exploit)</b>	<ul style="list-style-type: none"> <li>- Avoid 의 반대 개념, 리스크와 관련된 불확실성을 제거하여 기회가 확실히 일어날 수 있도록 하는 것</li> </ul>
	<b>공유 (Share)</b>	<ul style="list-style-type: none"> <li>- Transfer 의 반대 개념, 협력관계, 제휴 등의 예</li> <li>- 제 3 자와 제휴하여 이익을 공유함에 동시에 책임을 가지도록 하는 것.</li> </ul>
	<b>증대 (Enhanc)</b>	<ul style="list-style-type: none"> <li>- 긍정적 영향을 미치는 리스크의 주요 요인을 식별하여 기회의 규모를 더 키우는 것</li> </ul>
	<b>수용 (Accept)</b>	<ul style="list-style-type: none"> <li>- 리스크가 막 발생하려 하거나, 발생한 경우 어떤 식으로 대처하겠다는 비상계획을 개발해 놓는 것</li> </ul>

- 기존 위험의 심각성, 비용의 효과성, 시간적인 적절성, 프로젝트 상황에서의 현실성, 모든 관련된 그룹의 합의, 책임있는 개인/그룹에 할당 등을 고려하여 프로젝트 목표에 대한 기회는 늘리고, 위협은 감소시키기 위한 대안을 개발하고 행동해야 함

"끝"

#### 기출풀이의견

1. 정량적 위험분석 문제는 117회 정보관리기술사 문제와 정확하게 일치하는 문제임.
2. 각 위험분석 기법들에 대한 그림과 계산 방식이 정확해야 고득점 가능함.

## 6. 다음에 대하여 설명하시오.

가. 최단경로 알고리즘의 유형 4 가지

문제 제 나. 다음 그래프에서 4 가지 알고리즘 계산 방법

- A, B, C, D, E 5 개 노드로 구성된 그래프(A 가 출발지, E 가 목적지임)

다. "나"의 4 가지 알고리즘 계산결과 비교

출제영역	알고리즘	난이도	★★★★★
출제배경	최근 확률, 통계, 알고리즘 도메인에 대한 문제 빈번히 출제되고, 다익스트라 알고리즘 경우 현실 세계에 사용하기 매우 적합하여 출제		
출제빈도	미출제		
참고자료	<ul style="list-style-type: none"> <li>- 쉽게 배우는 알고리즘 (숙명여자대학교)</li> <li>- 최단 경로 탐색 A* 알고리즘 (GIS Developer)</li> <li>- 다익스트라 알고리즘 2절 단일출발점 최단경로</li> <li>- 알고리즘 도감(이시다 모리테루)</li> </ul>		
Keyword	음의간선, 2 차원 배열, 휴리스틱, 네비게이션, GPS, 각 알고리즘 개념 및 계산 방법		
풀이	김준삼(118 회 정보관리기술사)		
감수	공수재(111 회 컴퓨터시스템응용기술사)		

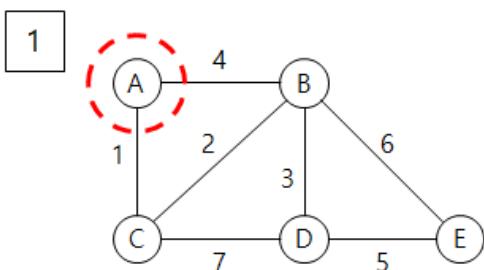
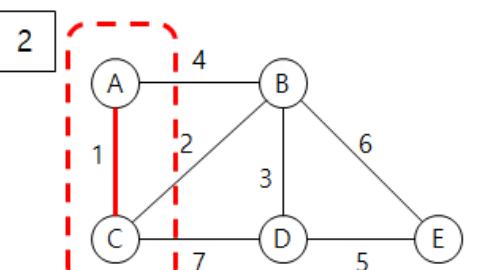
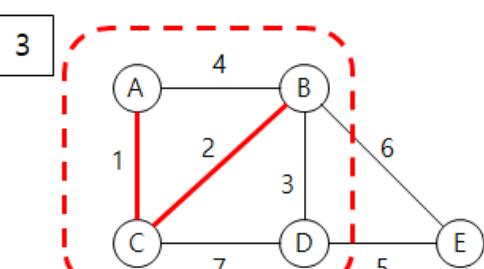
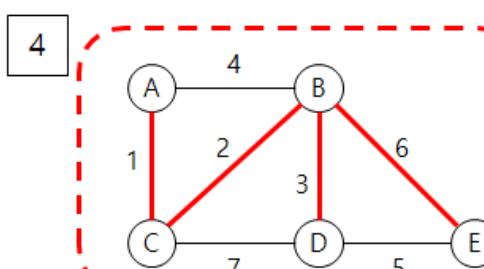
## 1. 최단경로 알고리즘의 유형 4 가지

구분	설명
다익스트라	- 하나의 시작 정점으로부터 모든 다른 정점까지의 최단 경로를 찾는 알고리즘
벨만 포드	- 하나의 시작 정점으로부터 모든 다른 정점까지 음의 간선도 해결 가능한 최단 경로 알고리즘
플로이드 와샬	- 2 차원 배열을 통해 모든 정점에서 모든 정점의 최단경로 문제를 해결하는 알고리즘
A* 알고리즘	- 탐색속도를 높이기 위해 휴리스틱 방법을 사용하여 최단경로 문제를 해결하는 알고리즘

- 최단 경로 문제는 정점과 정점을 연결하는 경로 중 간선들의 가중치 합이 최소가 되는 경로를 찾는 문제임

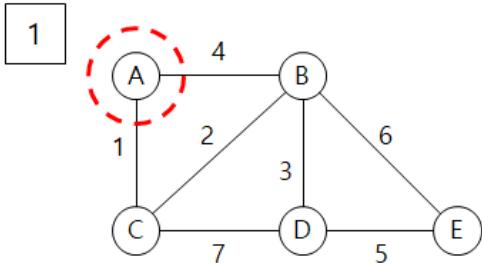
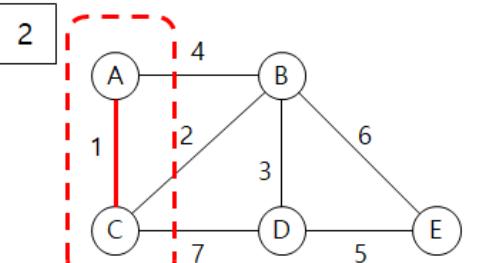
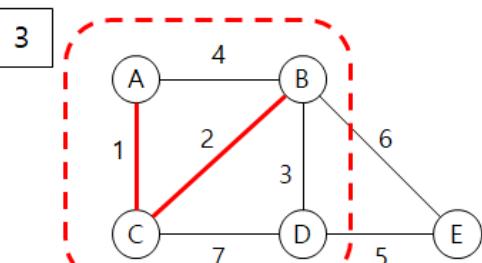
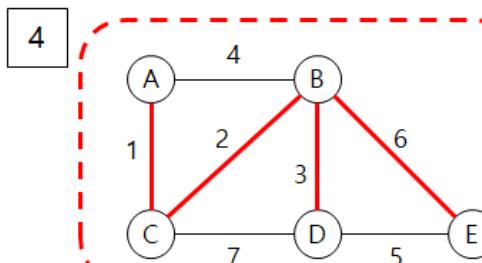
## 2. 다익스트라 및 벨만포드 알고리즘 계산방법

### 가. 다익스트라 알고리즘 계산방법

구 분	설 명																																																
개념	- 하나의 시작점에서 하나의 도착점으로 가는 최단경로 문제를 해결하는 알고리즘																																																
계산방법	<p>1 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td><math>\infty</math></td><td><math>\infty</math></td><td><math>\infty</math></td><td><math>\infty</math></td></tr> </tbody> </table> <p>2 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td>4</td><td><math>\infty</math></td><td><math>\infty</math></td><td><math>\infty</math></td></tr> </tbody> </table> <p>3 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td>3</td><td>1</td><td>7</td><td><math>\infty</math></td></tr> </tbody> </table> <p>4 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td>3</td><td>1</td><td>6</td><td>9</td></tr> </tbody> </table>	노드	A	B	C	D	E	비용	0	$\infty$	$\infty$	$\infty$	$\infty$	노드	A	B	C	D	E	비용	0	4	$\infty$	$\infty$	$\infty$	노드	A	B	C	D	E	비용	0	3	1	7	$\infty$	노드	A	B	C	D	E	비용	0	3	1	6	9
노드	A	B	C	D	E																																												
비용	0	$\infty$	$\infty$	$\infty$	$\infty$																																												
노드	A	B	C	D	E																																												
비용	0	4	$\infty$	$\infty$	$\infty$																																												
노드	A	B	C	D	E																																												
비용	0	3	1	7	$\infty$																																												
노드	A	B	C	D	E																																												
비용	0	3	1	6	9																																												
계산절차	<ol style="list-style-type: none"> <li>모든 정점 <math>\infty</math>(무한대)로 초기화 : 모든 정점 위에 있는 경로의 길이를 무한대로 초기화</li> <li>시작 정점 최단 경로 추가 : 시작 정점 경로 길이를 0 으로 초기화 후 최단 경로에 추가</li> <li>인접 정점 탐색 및 가중치 갱신 : 인접 정점의 경로길이를 갱신하고 이들을 최단 경로에 추가</li> <li>가중치 변화 없을 때까지 반복 : 그래프 내의 모든 정점이 최단경로에 소속될 때 까지 탐색 반복</li> </ol>																																																

- 다익스트라는 흔히 인공위성 GPS 소프트웨어에 가장 많이 사용되고, 현실 세계에 사용하기 매우 적합한 알고리즘 중 하나임.
- 단, 음의 간선이 포함될 경우 다익스트라 알고리즘 사용 불가하여, 벨만포드 알고리즘으로 계산함

## 나. 벨만포드 알고리즘 계산방법

구 분	설 명																																																
개념	<p>- 하나의 시작점에서 하나의 도착점으로 가는 음의 간선도 가능한 최단경로 문제를 해결하는 알고리즘</p>																																																
계산방법	<p>1 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td><math>\infty</math></td><td><math>\infty</math></td><td><math>\infty</math></td><td><math>\infty</math></td></tr> </tbody> </table> <p>2 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td>4</td><td>1</td><td><math>\infty</math></td><td><math>\infty</math></td></tr> </tbody> </table> <p>3 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td>3</td><td>1</td><td>8</td><td><math>\infty</math></td></tr> </tbody> </table> <p>4 </p> <table border="1"> <thead> <tr> <th>노드</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> </thead> <tbody> <tr> <td>비용</td><td>0</td><td>3</td><td>1</td><td>6</td><td>9</td></tr> </tbody> </table>	노드	A	B	C	D	E	비용	0	$\infty$	$\infty$	$\infty$	$\infty$	노드	A	B	C	D	E	비용	0	4	1	$\infty$	$\infty$	노드	A	B	C	D	E	비용	0	3	1	8	$\infty$	노드	A	B	C	D	E	비용	0	3	1	6	9
노드	A	B	C	D	E																																												
비용	0	$\infty$	$\infty$	$\infty$	$\infty$																																												
노드	A	B	C	D	E																																												
비용	0	4	1	$\infty$	$\infty$																																												
노드	A	B	C	D	E																																												
비용	0	3	1	8	$\infty$																																												
노드	A	B	C	D	E																																												
비용	0	3	1	6	9																																												
계산절차	<ol style="list-style-type: none"> <li>1) 시작 정점을 결정한다.</li> <li>2) 시작 정점부터 다른 정점까지 거리 값 모두 무한대로 초기화한다. (시작 정점은 0으로 초기화)</li> <li>3) 현재 정점의 모든 인접 정점들을 탐색하며, 기존에 기록된 인접 정점까지의 거리보다 현재 정점을 거쳐 인접 정점에 도달하는 거리가 더 짧다면 인접 정점까지의 거리를 갱신한다.</li> <li>4) 3 번 과정을 <math>V-1</math> 번 반복한다.</li> </ol>																																																

- 벨만포드의 경우 음수의 간선이 아닌 음수 순환이 도착점 전에 존재할 경우 문제 발생

- vertex 가 추가되면 연결 가능한 모든 간선을 계산해서 가중치를 update 함

- 단, 벨만포드는 시간 복잡도  $O(VE)$ 로 다익스트라 알고리즘 보다 다소 느리다는 단점 존재.

### 3. 플로이드 와샬 및 A\* 알고리즘 계산방법

#### 가. 플로이드 와샬 알고리즘 계산방법

구 분	설 명																																																																												
개념	- 2 차원 배열을 통해 모든 정점에서 모든 정점의 최단경로 문제를 해결하는 알고리즘																																																																												
계산방법	<b>1</b> 초기값 설정 <table border="1"> <tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> <tr><th>A</th><td>0</td><td>4</td><td>1</td><td><math>\infty</math></td><td><math>\infty</math></td></tr> <tr><th>B</th><td>4</td><td>0</td><td>2</td><td>3</td><td>6</td></tr> <tr><th>C</th><td>1</td><td>2</td><td>0</td><td>7</td><td><math>\infty</math></td></tr> <tr><th>D</th><td><math>\infty</math></td><td>3</td><td>7</td><td>0</td><td>5</td></tr> <tr><th>E</th><td><math>\infty</math></td><td>6</td><td><math>\infty</math></td><td>5</td><td>0</td></tr> </table>				A	B	C	D	E	A	0	4	1	$\infty$	$\infty$	B	4	0	2	3	6	C	1	2	0	7	$\infty$	D	$\infty$	3	7	0	5	E	$\infty$	6	$\infty$	5	0	<b>2</b> 노드 A 를 거쳐가는 경우 <table border="1"> <tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> <tr><th>A</th><td>0</td><td>4</td><td>1</td><td><math>\infty</math></td><td><math>\infty</math></td></tr> <tr><th>B</th><td>4</td><td>0</td><td>2</td><td>3</td><td>6</td></tr> <tr><th>C</th><td>1</td><td>2</td><td>0</td><td>7</td><td><math>\infty</math></td></tr> <tr><th>D</th><td><math>\infty</math></td><td>3</td><td>7</td><td>0</td><td>5</td></tr> <tr><th>E</th><td><math>\infty</math></td><td>6</td><td><math>\infty</math></td><td>5</td><td>0</td></tr> </table>			A	B	C	D	E	A	0	4	1	$\infty$	$\infty$	B	4	0	2	3	6	C	1	2	0	7	$\infty$	D	$\infty$	3	7	0	5	E	$\infty$	6	$\infty$	5	0
	A	B	C	D	E																																																																								
A	0	4	1	$\infty$	$\infty$																																																																								
B	4	0	2	3	6																																																																								
C	1	2	0	7	$\infty$																																																																								
D	$\infty$	3	7	0	5																																																																								
E	$\infty$	6	$\infty$	5	0																																																																								
	A	B	C	D	E																																																																								
A	0	4	1	$\infty$	$\infty$																																																																								
B	4	0	2	3	6																																																																								
C	1	2	0	7	$\infty$																																																																								
D	$\infty$	3	7	0	5																																																																								
E	$\infty$	6	$\infty$	5	0																																																																								
<b>3</b> 노드 B 를 거쳐가는 경우 <table border="1"> <tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> <tr><th>A</th><td>0</td><td>4</td><td>1</td><td><u>7</u></td><td><u>10</u></td></tr> <tr><th>B</th><td>4</td><td>0</td><td>2</td><td>3</td><td>6</td></tr> <tr><th>C</th><td>1</td><td>2</td><td>0</td><td>7</td><td><u>8</u></td></tr> <tr><th>D</th><td><u>7</u></td><td>3</td><td><u>5</u></td><td>0</td><td>5</td></tr> <tr><th>E</th><td><u>10</u></td><td>6</td><td><u>8</u></td><td>5</td><td>0</td></tr> </table>				A	B	C	D	E	A	0	4	1	<u>7</u>	<u>10</u>	B	4	0	2	3	6	C	1	2	0	7	<u>8</u>	D	<u>7</u>	3	<u>5</u>	0	5	E	<u>10</u>	6	<u>8</u>	5	0	<b>4</b> 노드 C 를 거쳐가는 경우 <table border="1"> <tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> <tr><th>A</th><td>0</td><td><u>3</u></td><td>1</td><td>7</td><td><u>9</u></td></tr> <tr><th>B</th><td><u>3</u></td><td>0</td><td>2</td><td>3</td><td>6</td></tr> <tr><th>C</th><td>1</td><td>2</td><td>0</td><td>7</td><td>8</td></tr> <tr><th>D</th><td><u>6</u></td><td>3</td><td>5</td><td>0</td><td>5</td></tr> <tr><th>E</th><td><u>9</u></td><td>6</td><td>8</td><td>5</td><td>0</td></tr> </table>			A	B	C	D	E	A	0	<u>3</u>	1	7	<u>9</u>	B	<u>3</u>	0	2	3	6	C	1	2	0	7	8	D	<u>6</u>	3	5	0	5	E	<u>9</u>	6	8	5	0	
	A	B	C	D	E																																																																								
A	0	4	1	<u>7</u>	<u>10</u>																																																																								
B	4	0	2	3	6																																																																								
C	1	2	0	7	<u>8</u>																																																																								
D	<u>7</u>	3	<u>5</u>	0	5																																																																								
E	<u>10</u>	6	<u>8</u>	5	0																																																																								
	A	B	C	D	E																																																																								
A	0	<u>3</u>	1	7	<u>9</u>																																																																								
B	<u>3</u>	0	2	3	6																																																																								
C	1	2	0	7	8																																																																								
D	<u>6</u>	3	5	0	5																																																																								
E	<u>9</u>	6	8	5	0																																																																								
<b>5</b> 노드 D 를 거쳐가는 경우 <table border="1"> <tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> <tr><th>A</th><td>0</td><td>3</td><td>1</td><td>7</td><td>9</td></tr> <tr><th>B</th><td>3</td><td>0</td><td>2</td><td>3</td><td>6</td></tr> <tr><th>C</th><td>1</td><td>2</td><td>0</td><td>7</td><td>8</td></tr> <tr><th>D</th><td>6</td><td>3</td><td>5</td><td>0</td><td>5</td></tr> <tr><th>E</th><td>9</td><td>6</td><td>8</td><td>5</td><td>0</td></tr> </table>				A	B	C	D	E	A	0	3	1	7	9	B	3	0	2	3	6	C	1	2	0	7	8	D	6	3	5	0	5	E	9	6	8	5	0	<b>6</b> 노드 E 를 거쳐가는 경우 <table border="1"> <tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr> <tr><th>A</th><td>0</td><td>3</td><td>1</td><td>7</td><td>9</td></tr> <tr><th>B</th><td>3</td><td>0</td><td>2</td><td>3</td><td>6</td></tr> <tr><th>C</th><td>1</td><td>2</td><td>0</td><td>7</td><td>8</td></tr> <tr><th>D</th><td>6</td><td>3</td><td>5</td><td>0</td><td>5</td></tr> <tr><th>E</th><td>9</td><td>6</td><td>8</td><td>5</td><td>0</td></tr> </table>			A	B	C	D	E	A	0	3	1	7	9	B	3	0	2	3	6	C	1	2	0	7	8	D	6	3	5	0	5	E	9	6	8	5	0	
	A	B	C	D	E																																																																								
A	0	3	1	7	9																																																																								
B	3	0	2	3	6																																																																								
C	1	2	0	7	8																																																																								
D	6	3	5	0	5																																																																								
E	9	6	8	5	0																																																																								
	A	B	C	D	E																																																																								
A	0	3	1	7	9																																																																								
B	3	0	2	3	6																																																																								
C	1	2	0	7	8																																																																								
D	6	3	5	0	5																																																																								
E	9	6	8	5	0																																																																								
계산절차	1) <u>초기 2 차원 배열 구성</u> : 모든 정점간의 가중치를 갱신 2) <u>인접 정점 탐색 및 가중치 갱신</u> : 인접 정점간의 최소 가중치로 갱신 3) <u>경유하는 정점 탐색 및 가중치 갱신</u> : 목적지 정점으로 경유하는 비용이 최소값이면 갱신 4) <u>가중치 변화 없을 때까지 반복</u> : 모든 정점간에 가중치 변화가 없을 때까지 반복																																																																												

- 플로이드와샬 알고리즘은 모든 정점과 정점 사이의 최단거리를 구할 수 있는 것이 가장 큰 특징

## 나. A\* 알고리즘 계산방법

구 분	설 명																																																																																															
개념	- 탐색속도를 높이기 위해 휴리스틱 방법을 사용하여 최단경로 문제를 해결하는 알고리즘																																																																																															
특징	- 휴리스틱 추정값을 어떤 방식으로 설정하느냐에 따라 얼마나 빨리 최단 경로를 파악할 수 있느냐가 결정됨.																																																																																															
계산방법	<p>1 시작점 A 인접 노드 계산</p> <p>O = <table border="1"> <thead> <tr> <th>Node ID</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>F Score</td> <td>10</td> <td>13</td> </tr> <tr> <td>G Score</td> <td>4</td> <td>1</td> </tr> <tr> <td>H Score</td> <td>6</td> <td>12</td> </tr> <tr> <td>Parent Node</td> <td>A</td> <td>A</td> </tr> </tbody> </table></p> <p>- O (Open List) : 최단 경로를 분석하기 위한 상태값들이 계속 갱신  - C (Close List) : 처리가 완료된 노드를 담아 두기 위한 목적</p> <p>C = <table border="1"> <thead> <tr> <th>Node ID</th> <th>A</th> </tr> </thead> <tbody> <tr> <td>F Score</td> <td>0</td> </tr> <tr> <td>G Score</td> <td>0</td> </tr> <tr> <td>H Score</td> <td>0</td> </tr> <tr> <td>Parent Node</td> <td>-</td> </tr> </tbody> </table></p> <p>2 B 인접 노드 C, D 노드 추가      3 도착점 결정</p> <p>O = <table border="1"> <thead> <tr> <th>Node ID</th> <th>C</th> <th>D</th> <th>E</th> </tr> </thead> <tbody> <tr> <td>F Score</td> <td>13</td> <td>12</td> <td>10</td> </tr> <tr> <td>G Score</td> <td>1</td> <td>7</td> <td>10</td> </tr> <tr> <td>H Score</td> <td>12</td> <td>5</td> <td>0</td> </tr> <tr> <td>Parent Node</td> <td>A</td> <td>B</td> <td>B</td> </tr> </tbody> </table></p> <p>C = <table border="1"> <thead> <tr> <th>Node ID</th> <th>A</th> <th>B</th> </tr> </thead> <tbody> <tr> <td>F Score</td> <td>0</td> <td>10</td> </tr> <tr> <td>G Score</td> <td>0</td> <td>4</td> </tr> <tr> <td>H Score</td> <td>0</td> <td>6</td> </tr> <tr> <td>Parent Node</td> <td>-</td> <td>A</td> </tr> </tbody> </table></p> <p>O = <table border="1"> <thead> <tr> <th>Node ID</th> <th>C</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>F Score</td> <td>13</td> <td>12</td> </tr> <tr> <td>G Score</td> <td>1</td> <td>7</td> </tr> <tr> <td>H Score</td> <td>12</td> <td>5</td> </tr> <tr> <td>Parent Node</td> <td>A</td> <td>B</td> </tr> </tbody> </table></p> <p>C = <table border="1"> <thead> <tr> <th>Node ID</th> <th>A</th> <th>B</th> <th>E</th> </tr> </thead> <tbody> <tr> <td>F Score</td> <td>0</td> <td>10</td> <td>10</td> </tr> <tr> <td>G Score</td> <td>0</td> <td>4</td> <td>10</td> </tr> <tr> <td>H Score</td> <td>0</td> <td>6</td> <td>0</td> </tr> <tr> <td>Parent Node</td> <td>-</td> <td>A</td> <td>B</td> </tr> </tbody> </table></p> <ul style="list-style-type: none"> <li>- F = G + H</li> <li>- G : 시작 노드에서 해당 노드까지의 실제 소요 경비값</li> <li>- H : 휴리스틱 추정값으로 해당 노드에서 최종 목적지까지 도달하는데 소요될 것이라고 추정되는 값</li> <li>- Parent Node: 해당 노드에 도달하기 직전에 거치는 노드 번호</li> </ul>	Node ID	B	C	F Score	10	13	G Score	4	1	H Score	6	12	Parent Node	A	A	Node ID	A	F Score	0	G Score	0	H Score	0	Parent Node	-	Node ID	C	D	E	F Score	13	12	10	G Score	1	7	10	H Score	12	5	0	Parent Node	A	B	B	Node ID	A	B	F Score	0	10	G Score	0	4	H Score	0	6	Parent Node	-	A	Node ID	C	D	F Score	13	12	G Score	1	7	H Score	12	5	Parent Node	A	B	Node ID	A	B	E	F Score	0	10	10	G Score	0	4	10	H Score	0	6	0	Parent Node	-	A	B
Node ID	B	C																																																																																														
F Score	10	13																																																																																														
G Score	4	1																																																																																														
H Score	6	12																																																																																														
Parent Node	A	A																																																																																														
Node ID	A																																																																																															
F Score	0																																																																																															
G Score	0																																																																																															
H Score	0																																																																																															
Parent Node	-																																																																																															
Node ID	C	D	E																																																																																													
F Score	13	12	10																																																																																													
G Score	1	7	10																																																																																													
H Score	12	5	0																																																																																													
Parent Node	A	B	B																																																																																													
Node ID	A	B																																																																																														
F Score	0	10																																																																																														
G Score	0	4																																																																																														
H Score	0	6																																																																																														
Parent Node	-	A																																																																																														
Node ID	C	D																																																																																														
F Score	13	12																																																																																														
G Score	1	7																																																																																														
H Score	12	5																																																																																														
Parent Node	A	B																																																																																														
Node ID	A	B	E																																																																																													
F Score	0	10	10																																																																																													
G Score	0	4	10																																																																																													
H Score	0	6	0																																																																																													
Parent Node	-	A	B																																																																																													
결과	<ul style="list-style-type: none"> <li>- 목적지의 E 노드에서 Parent Node 는 B 노드이고, B 노드의 Parent Node 는 A 이므로</li> <li>- 최단경로는 A -&gt; B -&gt; E 노드 설정됨</li> </ul>																																																																																															

- A\* 알고리즘은 시작 노드만을 지정해 다른 모든 노드에 대한 최단 경로를 파악하는 다익스트라 알고리즘과 다르게 시작 노드와 목적지 노드를 분명하게 지정해 이 두 노드 간의 최단 경로 계산 가능함

#### 4. 4 가지 알고리즘 계산결과 비교

구 분	다익스트라	벨만포드	플로이드 와샬	A*
최단거리	9	9	9	10
경로탐색	단일 정점 계산	단일 정점 계산	모든 정점 계산	단일 정점 출발
복잡도	$O(V+E)\log V$	$O(VE)$	$O(V^3)$	휴리스틱 추정값에 따라 변경
특징	음의 간선 계산 불가	음의 간선 계산 가능	2 차원 배열 모든 노드 경로 계산	휴리스틱 기반
활용	링크상태 라우팅	거리벡터 라우팅	GIS, 네트워크 분석	네비게이션

- 최단 경로 알고리즘은 특성에 따라 네이게이션, GIS 소프트웨어, 네트워크 라우팅 등에 사용되므로, 특성을 분석하여 최적의 알고리즘 선택 필요함.



#### 기출풀이 의견

1. 최단거리 4개의 계산 방법에 대한 질문으로 대부분 선택하지 않았을 것이라 예상함
2. 계산 문제이기 때문에 정답 및 답안작성 템플릿도 구성하기 어려워, 정확하게 작성시 20점 이상 예상